Universitatea Tehnică ”Gheorghe Asachi” din Iași
Facultatea de Automatică și Calculatoare
Domeniul Calculatoare și Tehnologia Informației
Specializarea Tehnologia Informației

Proiectarea Bazelor de Date

Sistem de Gestiune a unei Biblioteci

Student:
Martin Maria, 1410A

Cadru didactic coordonator:
Asoc. Ing. Baltariu Ionuț-Alexandru

An universitar 2023-2024

Cuprins

1. Descrierea Proiectului

Proiectul realizat în PL/SQL urmărește gestionarea eficientă a unei biblioteci. Prin utilizarea tabelelor, a legăturilor dintre acestea și a constrângerilor, aplicația asigură o evidență coerentă a împrumuturilor de cărți din bibliotecă. Tabela *carti* ține evidența împrumuturilor unei cărți la momentul actual. Tabela *detalii_carte* oferă informații esențiale despre fiecare carte. Tabela *bibliotecari* conține numele fiecărui bibliotecar, iar în tabela *membri* sunt incluse detalii importante despre persoanele care împrumută cărți. În tabela *imprumuturi*, se ține evidența cărților împrumutate, a persoanelor care le-au împrumutat, a bibliotecarilor care au realizat împrumutul, a zilei împrumutului și a datei de returnare a cărții. Pachetele din proiect facilitează operațiile de inserare, ștergere, actualizare și vizualizare a datelor din tabele, asigurând astfel gestionarea informațiilor din bibliotecă.

2. Teste

Pentru a verifica corectitudinea operațiilor și funcționalitatea proiectului și pentru a evita situațiile neprevăzute, am realizat câteva teste. Acestea asigură că aplicația funcționează conform cerințelor, iar operațiile executate respectă toate constrângerile impuse.

- **Testare funcție pentru afișarea cărților după gen**

Pentru a verifica funcția *get_carti_dupa_gen*, am afișat toate cărțile de tipul Fantasy.

```
Gen carti: Fantasy
Titlu: The Hobbit, Autor: J.R.R. Tolkien, Data Publicarii: 21-SEP-2037
Titlu: Harry Potter and the Sorcerer's Stone, Autor: J.K. Rowling, Data Publicarii: 26-JUN-2097
Titlu: A Game of Thrones, Autor: George R.R. Martin, Data Publicarii: 06-AUG-2096
Titlu: The Name of the Wind, Autor: Patrick Rothfuss, Data Publicarii: 27-MAR-2007
Titlu: The Fellowship of the Ring, Autor: J.R.R. Tolkien, Data Publicarii: 29-JUL-2054
Afisare completa
```

Figura 2.1 Afișare cărți după gen

- **Testare funcție pentru afișarea cărților după autor**

Pentru a verifica funcția *get_carti_dupa_autor*, am afișat toate cărțile autorului J.R.R. Tolkien.

```
Autor: J.R.R. Tolkien
Titlu: The Hobbit, Gen: Fantasy, Data Publicarii: 21-SEP-2037
Titlu: The Fellowship of the Ring, Gen: Fantasy, Data Publicarii: 29-JUL-2054
Afisare completa.
```

Figura 2.2 Afișare cărți după autor

- **Testare funcție pentru afișarea cărților disponibile**

Pentru a verifica funcția *get_carti_disponibile*, am afișat toate cărțile disponibile pentru împrumut.

```
Carti disponibile:
Titlu: The Name of the Wind, Autor: Patrick Rothfuss, Gen: Fantasy
Titlu: Pride and Prejudice, Autor: Jane Austen, Gen: Romance
Titlu: Outlander, Autor: Diana Gabaldon, Gen: Romance
Titlu: The Notebook, Autor: Nicholas Sparks, Gen: Romance
Titlu: Big Little Lies, Autor: Liane Moriarty, Gen: Mistery
Titlu: Treasure Island, Autor: Robert Louis Stevenson, Gen: Adventure
Titlu: Around the World in Eighty Days, Autor: Jules Verne, Gen: Adventure
Titlu: Frankenstein, Autor: Mary Shelley, Gen: Horror
Afisare completa.
```

Figura 2.3 Afișare cărți disponibile

- **Testare pachet pentru inserare**

Pentru a verifica funcțiile din pachetul *insert_pkg*, am inserat o nouă valoare în fiecare tabel și mesajul de succes. Am testat funcțiile *adauga_carte*, *adauga_detalii_carte*, *adauga_membru*, *adauga_bibliotecar* și *adauga_imprumut*. De asemenea, am verificat declanșarea trigger-ului la adăugarea unui împrumut.

```
Carte adaugata cu succes


PL/SQL procedure successfully completed.

Detalii carte adaugate cu succes


PL/SQL procedure successfully completed.

Membru adaugat cu succes


PL/SQL procedure successfully completed.

Bibliotecar adaugat cu succes


PL/SQL procedure successfully completed.

Membrul Andrei Popescu nu a returnat cartea "The Girl with the Dragon Tattoo" la timp.
Numar de zile de intarziere: 500
Imprumut adaugat cu succes
```

Figura 2.4 Adăugare date în tabele + trigger declanșat de adăugarea împrumutului

- **Testare pachet actualizare**

Pentru a verifica procedurile din pachetul *update_pkg*, am actualizat câte o valoare din fiecare tabel. Am testat funcțiile *update_carte*, *update_detalii_carte*, *update_membru*, *update_bibliotecar* și *update_imprumut*. De asemenea, am verificat declanșarea trigger-ului la actualizarea unui împrumut.

```
PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

Membrul Andrei Gigi nu a returnat cartea "The Girl with the Dragon Tattoo" la timp.
Numar de zile de intarziere: 500


PL/SQL procedure successfully completed.
```

Figura 2.5 Actualizare tabele + trigger declanșat de actualizarea împrumutului

- **Testare pachet ștergere**

Pentru a verifica procedurile din pachetul delete_pkg, am testat ștergerea unui element dintr-o tabelă atât cu cascade constraint, cât și fără, pentru a verifica diferite scenarii. Am testat funcțiile *delete_carte* cu cascade constraint setat la TRUE, *delete_detalii_carte*, *delete_membru* cu cascade constraint setat la TRUE, *delete_bibliotecar* cu cascade constraint setat la FALSE și *delete_imprumut*.

```
PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.


Error starting at line : 123 in command -
BEGIN
    delete_pkg.delete_bibliotecar(2, FALSE);
END;
Error report -
ORA-20001: Nu se poate sterge bibliotecarul, exista imprumuturi asociate.
ORA-06512: at "PBD_STUD_PROJ_124.DELETE_PKG", line 39
ORA-06512: at "PBD_STUD_PROJ_124.DELETE_PKG", line 22
ORA-06512: at line 2


PL/SQL procedure successfully completed.
```

Figura 2.6 Ștergere date din tabele

- **Testare pachet vizualizări**

Pentru a verifica procedurile din pachetul *view_pkg*, am afișat împrumuturile asociate membrului cu ID-ul 10, împrumuturile asociate cărții cu ID-ul 11, detaliile cărții cu ID-ul 5 și împrumuturile realizate de către bibliotecarul cu ID-ul 4. Am testat funcțiile *view_imprumuturi_by_membru*, *view_imprumuturi_by_carte*, *view_detalii_carte_by_carte* și *view_imprumuturi_by_bibliotecar*.

```
Imprumut ID: 7, Carte ID: 8, Titlu: The Notebook, Autor: Nicholas Sparks, Data Imprumut: 11-APR-24, Data Returnare: 21-APR-24, Bibliotecar ID: 4
Imprumut ID: 5, Carte ID: 9, Titlu: Me Before You, Autor: Jojo Moyes, Data Imprumut: 13-SEP-18, Data Returnare: 23-SEP-18, Bibliotecar ID: 2
Imprumut ID: 9, Carte ID: 17, Titlu: Dracula, Autor: Bram Stoker, Data Imprumut: 11-JAN-24, Data Returnare: 21-JAN-24, Bibliotecar ID: 1


PL/SQL procedure successfully completed.

Imprumut ID: 6, Carte ID: 11, Membru: Maria Ionescu, Adresa: Strada Primaverii, Nr. 22, Data Imprumut: 15-JUN-10, Data Returnare: 25-JUN-10, Bibliotecar ID: 2


PL/SQL procedure successfully completed.

Detalii Carte ID: 5, Carte ID: 5, Titlu: The Fellowship of the Ring, Autor: J.R.R. Tolkien, Gen: Fantasy, Data Publicarii: 29-JUL-54


PL/SQL procedure successfully completed.

Imprumut ID: 7, Carte ID: 8, Membru: Simona Grigorescu, Adresa: Strada Gheorghe Doja, Nr. 17, Titlu: The Notebook, Autor: Nicholas Sparks, Data Imprumut: 11-APR-24, Data Returnare: 21-APR-24, Bibliotecar ID: 4
```

Figura 2.7 Vizualizare date din tabele

- **Testare cursor pentru verificarea și actualizarea împrumuturilor**

Pentru a verifica procedura *verifica_imprumuturi_si_actualizeaza* din *cursor_pkg*, am verificat și actualizat starea împrumuturilor în funcție de data returnării.

```
Imprumut ID: 1 - Status carte actualizat la: F (disponibil)
Imprumut ID: 2 - Status carte actualizat la: F (disponibil)
Imprumut ID: 4 - Status carte actualizat la: F (disponibil)
Imprumut ID: 5 - Status carte actualizat la: F (disponibil)
Imprumut ID: 6 - Status carte actualizat la: F (disponibil)
Imprumut ID: 7 - Status carte actualizat la: F (disponibil)
Imprumut ID: 9 - Status carte actualizat la: F (disponibil)
Imprumut ID: 11 - Status carte actualizat la: T (imprumutat)
Imprumut ID: 12 - Status carte actualizat la: F (disponibil)
Imprumut ID: 13 - Status carte actualizat la: F (disponibil)


PL/SQL procedure successfully completed.
```

Figura 2.8 Verificare și actualizare împrumuturi prin cursor

- **Testare tranzacție pentru inserare carte și detalii carte**

Pentru a verifica integritatea tranzacțiilor, am inserat o carte și detaliile acesteia în cadrul unei singure tranzacții, deja existente, pentru a declanșa excepția ce face ROLLBACK.

```
Eroare la adaugarea cartii: ORA-00001: unique constraint (PBD_STUD_PROJ_124.CARTI_PK) violated
Eroare la adaugarea detaliilor cartii: ORA-00001: unique constraint (PBD_STUD_PROJ_124.DETALII_CARTE__IDX) violated


PL/SQL procedure successfully completed.
```

Figura 2.9 Tranzacție pentru inserare carte și detalii carte

- **Testare tranzacție pentru inserare membru și bibliotecar**

Pentru a verifica integritatea tranzacțiilor, am inserat un membru și un bibliotecar în cadrul unei singure tranzacții, deja existenți, pentru a declanșa excepția ce face ROLLBACK.

```
Eroare la adaugarea membrului: ORA-00001: unique constraint (PBD_STUD_PROJ_124.MEMBRI_PK) violated
Eroare la adaugarea bibliotecarului: ORA-00001: unique constraint (PBD_STUD_PROJ_124.BIBLIOTECARI_PK) violated


PL/SQL procedure successfully completed.
```

Figura 2.10 Tranzacție pentru inserare membru și bibliotecar

- **Testare pentru adăugarea unui împrumut pentru o carte care nu există**

Pentru a testa gestionarea erorilor la adăugarea unui împrumut, am creat un scenariu în care se încearcă adăugarea unui împrumut pentru o carte care nu există în baza de date. În acest test, funcția *adauga_imprumut* încearcă să adauge un împrumut pentru o carte inexistentă. Se așteaptă ca funcția să returneze un mesaj de eroare specific, confirmând că integritatea referințelor este menținută.

```
Eroare: Carte cu id-ul specificat nu exista
```

Figura 2.11 Adăugarea unui împrumut pentru o carte care nu există

- **Testare pentru adăugarea unui împrumut cu data de returnare mai mică decât data de împrumut**

Pentru a testa constrângerile asupra datelor, am creat un scenariu în care se încearcă inserarea unui împrumut cu o dată de returnare mai mică decât data de împrumut. Acest test verifică dacă constrângerea definită pe tabelul *imprumuturi* funcționează corect. Funcția *adauga_imprumut* încearcă să insereze un împrumut cu o dată de returnare invalidă. Se așteaptă ca constrângerea *imprumuturi_ck_1* să împiedice această inserare și să genereze o eroare, confirmând că datele respectă constrângerile de integritate definite.

```
Eroare la adaugarea imprumutului: ORA-02290: check constraint (PBD_STUD_PROJ_124.IMPRUMUTURI_CK_1) violated
```

Figura 2.12 Adăugarea unui împrumut cu data de returnare mai mică decât data de împrumut

3. Structura și Inter-relaționarea Tabelelor

3.1 Diagrama ER - modelul relațional

Modelul relațional al proiectului este structurat în jurul a cinci tabele principale, fiecare având un rol specific în gestionarea eficientă a unei biblioteci. Aceste tabele sunt interconectate prin diverse relații și constrângeri, asigurând integritatea datelor și facilitând operațiunile necesare pentru urmărirea împrumuturilor de cărți.

Tabela *carti* stochează informații despre cărți, incluzând ID-ul unic al fiecărei cărți și statusul acesteia, indicând dacă este disponibilă sau împrumutată. Tabela *detalii_carte* conține informații detaliate despre fiecare carte, cum ar fi titlul, autorul, genul și data publicării, fiind asociată direct cu tabelul *carti*. În tabelul *bibliotecari* sunt stocate informațiile despre bibliotecari, incluzând ID-ul și numele fiecărui bibliotecar. Tabela *membri* conține informații despre membrii bibliotecii, cum ar fi numele, adresa, numărul de telefon, emailul și CNP-ul fiecărui membru. Tabela *imprumuturi* ține evidența împrumuturilor de cărți, incluzând informații despre cartea împrumutată, membrul care a împrumutat cartea, bibliotecarul care a procesat împrumutul, data împrumutului și data returnării.

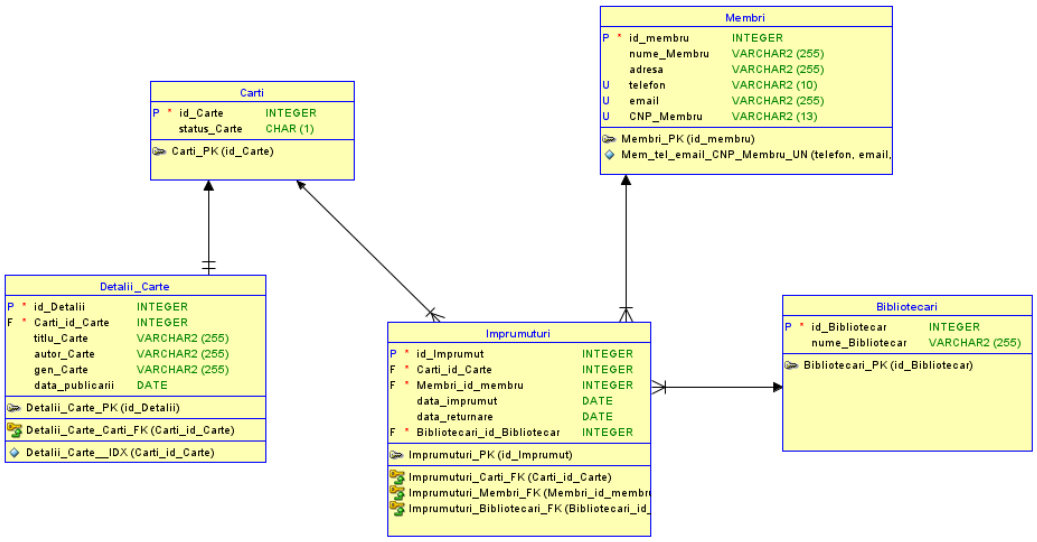

Figura 3.1 Diagrama ER

3.2 Relații între tabele

Pentru gestionarea bibliotecii, există mai multe tipuri de relații între tabele, fiecare jucând un rol important în asigurarea coerenței datelor:

- Relația **1:1** între tabela *carti* și tabela *detalii_carte*. Fiecare carte poate avea doar o înregistrare pentru detaliile asociate, iar fiecare înregistrare din *detalii_carte* este asociată cu o singură carte.
- Relația **1:N** între tabela *bibliotecari* și tabela *imprumuturi*. Fiecare bibliotecar poate gestiona mai multe împrumuturi, dar fiecare împrumut este asociat cu un singur bibliotecar.
- Relația **1:N** între tabela *membri* și tabela *imprumuturi*. Fiecare membru poate împrumuta mai multe cărți, dar fiecare împrumut este asociat cu un singur membru.
- Relația **1:N** între tabela *carti* și tabela *imprumuturi*. Fiecare carte poate fi împrumutată de mai multe ori, dar fiecare împrumut este asociat cu o singură carte.

3.3 Constrângeri

În cadrul proiectului, au fost utilizate diverse tipuri de constrângeri pentru a asigura validitatea și relațiile corecte între tabele.

- **Cheia Primară (Primary Key)**: Fiecare tabel are definită o cheie primară pentru a asigura unicitatea fiecărei înregistrări. De exemplu, tabela *carti* are *id_carte* ca și cheie primară, *detalii_carte* are *id_detalii*, *bibliotecari* are *id_bibliotecar*, *imprumuturi* are *id_imprumut*, iar *membri* are *id_membru*.
- **Cheia Străină (Foreign Key)**: Cheile străine sunt utilizate pentru a menține relațiile între tabele. De exemplu, *carti_id_carte* din tabela *detalii_carte* este o cheie străină referențiind *id_carte* din tabela *carti*. Similar, *bibliotecari_id_bibliotecar* din *imprumuturi* referențiază

id_bibliotecar din *bibliotecari*, *carti_id_carte* din *imprumuturi* referențiază *id_carte* din carti, iar *membri_id_membru* din *imprumuturi* referențiază *id_membru* din *membri*.

- **Constrângerile de Unicitate (Unique Constraints)**: Acestea asigură faptul că anumite coloane au valori unice în întreg tabelul. De exemplu, în tabela *membri*, coloanele *telefon*, *email* și *cnp_membru* are o constrângere de unicitate pentru a preveni introducerea de date duplicate.
- **Constrângerile de Verificare (Check Constraints)**: Sunt utilizate pentru a valida datele înainte de a fi stocate în tabel. De exemplu, în tabela *imprumuturi*, există o constrângere *imprumuturi_ck_1* care asigură că *data_imprumut* este mai mică decât *data_returnare*.

4. Descrierea Logicii Stocate

4.1 Pachete, Proceduri și Funcții

Pentru a facilita gestionarea operațiilor și vizualizarea datelor în baza de date a bibliotecii, proiectul utilizează mai multe pachete și funcții.

Pachetul *cursor_pkg* gestionează verificarea și actualizarea stării împrumuturilor. Procedura *verifica_imprumuturi_si_actualizeaza* verifică starea împrumuturilor și actualizează statusul cărților în funcție de data returnării acestora.

Pachetul *delete_pkg* gestionează operațiile de ștergere pentru diverse entități din baza de date. Procedura *delete_bibliotecar* șterge un bibliotecar, verificând dacă există împrumuturi asociate acestuia și oferind opțiunea de ștergere în cascadă. Procedura *delete_carte* șterge o carte și poate șterge și detaliile asociate acesteia dacă este specificat sau împrumuturile asociate, oferind opțiunea de ștergere în cascadă. Procedura *delete_detalii_carte* șterge detalii despre o carte specificată. Procedura *delete_membru* șterge un membru, verificând dacă există împrumuturi asociate și oferind opțiunea de ștergere în cascadă. Procedura *delete_imprumut* șterge un împrumut specificat.

Pachetul *insert_pkg* gestionează operațiile de inserare pentru diverse entități din baza de date. Funcția *adauga_carte* adaugă o nouă carte cu un anumit status în tabelul *carti*. Funcția *adauga_detalii_carte* adaugă detalii pentru o carte, precum titlu, autor, gen și data publicării. Funcția *adauga_membru* adaugă un nou membru în tabelul *membri*, având informații precum nume, adresă, telefon, email și CNP. Funcția *adauga_bibliotecar* adaugă un nou bibliotecar în tabelul *bibliotecari*. Funcția *adauga_imprumut* adaugă un nou împrumut, verificând existența cărții, membrului și bibliotecarului specificat.

Pachetul *update_pkg* gestionează operațiile de actualizare pentru diverse entități din baza de date. Procedura *update_bibliotecar* actualizează numele unui bibliotecar specificat. Procedura *update_carte* actualizează statusul unei cărți specificate. Procedura *update_detalii_carte* actualizează detaliile unei cărți, precum titlul, autorul, genul și data publicării. Procedura *update_membru* actualizează informațiile unui membru, precum numele, adresa, telefonul, emailul și CNP-ul. Procedura *update_imprumut* actualizează informațiile unui împrumut, precum cartea, membrul, datele împrumutului și bibliotecarul asociat.

Pachetul *view_pkg* furnizează funcționalități pentru vizualizarea datelor din baza de date. Procedura *view_imprumuturi_by_membru* afișează împrumuturile asociate unui membru specificat, incluzând detalii despre cărțile împrumutate și datele relevante. Procedura *view_imprumuturi_by_carte* afișează împrumuturile asociate unei cărți specifice, incluzând detalii despre membrii și bibliotecarii

implicați. Procedura *view_detalii_carte_by_carte* afișează detaliile asociate unei cărți specificate, incluzând titlul, autorul, genul și data publicării. Procedura *view_imprumuturi_by_bibliotecar* afișează împrumuturile gestionate de un bibliotecar specificat, incluzând detalii despre cărțile și membrii implicați.

Funcția *get_carti_disponibile* returnează un mesaj care listează toate cărțile disponibile pentru împrumut. Funcția parcurge tabelul *detalii_carte* pentru a obține detalii despre cărțile care au statusul de disponibil în tabelul *carti*.

Funcția *get_carti_dupa_autor* returnează o listă de cărți scrise de un anumit autor. Aceasta primește ca parametru numele autorului și parcurge tabelul *detalii_carte* pentru a extrage titlul, genul și data publicării cărților scrise de autorul respectiv.

Funcția *get_carti_dupa_gen* returnează o listă de cărți dintr-un anumit gen. Aceasta primește ca parametru genul cărților și parcurge tabelul *detalii_carte* pentru a extrage titlul, autorul și data publicării cărților care aparțin genului specificat.

4.2 Triggeri

```
Membrul Elena Vasilescu nu a returnat cartea "The Hobbit" la timp.
Numar de zile de intarziere: 287
Imprumut adaugat cu succes
Membrul Maria Ionescu nu a returnat cartea "Harry Potter and the Sorcerer's Stone" la timp.
Numar de zile de intarziere: 125
Imprumut adaugat cu succes
Membrul Mihai Petrescu nu a returnat cartea "A Game of Thrones" la timp.
Numar de zile de intarziere: 1081
Imprumut adaugat cu succes
Membrul Ioan Dumitrescu nu a returnat cartea "The Fellowship of the Ring" la timp.
Numar de zile de intarziere: 1564
Imprumut adaugat cu succes
Membrul Simona Grigorescu nu a returnat cartea "Me Before You" la timp.
Numar de zile de intarziere: 2071
Imprumut adaugat cu succes
```

Figura 4.2.1 Trigger declanșat de inserarea unui nou împrumut

Pentru gestionarea bibliotecii, am utilizat un trigger, *check_return*, definit pentru a fi declanșat după fiecare inserare sau actualizare în tabelul *imprumuturi*.

Acest trigger verifică dacă data returnării unei cărți a trecut și, dacă da, afișează un mesaj în care specifică numele membrului care nu a returnat cartea la timp, titlul cărții ce trebuie returnată și numărul zilelor de întârziere.

4.3 Gestionarea Excepțiilor și Cursorilor

Pachetul *cursor_pkg* gestionează verificarea și actualizarea împrumuturilor. Procedura *verifica_imprumuturi_si_actualizeaza* utilizează un cursor pentru a parcurge toate împrumuturile și a actualiza statusul cărților în funcție de data returnării. Dacă data returnării a trecut, statusul cărții este setat la "F" (disponibil), altfel la "T" (împrumutat). Procedura gestionează excepțiile prin rollback în caz de erori și afișează mesaje relevante pentru a indica problemele întâlnite.

Pachetul *delete_pkg* include proceduri pentru ștergerea diferitelor entități din baza de date. Aceste proceduri utilizează puncte de salvare și rollback pentru a asigura integritatea datelor în caz de erori, verificând de asemenea existența relațiilor care ar putea împiedica ștergerea entităților.

Pachetul *insert_pkg* gestionează adăugarea de noi înregistrări în baza de date. Funcțiile returnează un mesaj de succes sau de eroare și utilizează puncte de salvare și rollback pentru gestionarea excepțiilor.

Pachetul *update_pkg* se ocupă de actualizarea informațiilor existente. Procedurile utilizează puncte de salvare și rollback pentru a gestiona erorile și pentru a asigura consistența datelor.

Pachetul *view_pkg* oferă funcționalități de vizualizare a datelor. Procedurile utilizează bucle pentru a parcurge înregistrările și a afișa informațiile relevante.

Triggerul *check_return* este activat după fiecare inserare sau actualizare în tabelul imprumuturi. Acesta verifică dacă data returnării unei cărți a trecut și, dacă este cazul, afișează mesaje care indică membrul care nu a returnat cartea la timp și numărul de zile de întârziere. Triggerul gestionează excepțiile prin afișarea unor mesaje de eroare corespunzătoare, fără a modifica statusul cărții în baza de date.