Universitatea Tehnică „Gheorghe Asachi”, Iași

Facultatea de Automatică și Calculatoare

Specializarea: TI

Disciplina: Proiectarea Bazelor de Date

Gestionarea unei școli de balet

Coordonator: Ș.l.dr.ing. Mironeanu Cătălin

Student: Iorga Elena

Grupa: 1411B

1. Descrierea proiectului

Baza de date „Gestionarea unei școli de balet” are ca scop oferirea de suport pentru împărțirea studenților școlii de balet pe grupe, asignarea instructorilor și a pianiștilor grupei, organizarea spectacolelor și a asignării sălilor pentru repetiții. Este necesară o astfel de structură întrucât este greu să se țină o astfel de evidență fără un ajutor structurat.

Școala de balet curentă are la dispoziție 6 săli, instructori si pianiști care ii vor ajuta pe elevi sa evolueze, iar rezultatele lor se vor vedea atunci când participa la spectacole. La un spectacol nu pot participa mai mult de 30 de persoane.

Fiecare elev își va completa statutul la înscriere și apoi va fi asignat într-o grupă cu același statut ca el dacă este disponibilă. Dacă nu este disponibilă, înseamnă că școala este plină și se va afișa un mesaj astfel încât administratorii să știe dacă trebuie să adauge noi grupe și nou personal. Într-o grupa pot fi maxim câți elevi încap în sala asignată lor. Daca un elev care deja este profesionist are numărul suficient de spectacole, atunci el nu mai poate avansa. Daca se încearcă avansarea lui, acesta va rămâne exact unde este, doar spectacolele la care a participat vor fi adăugate în istoric.

Spectacolele pot fi adăugate în baza de date după ce au avut loc pentru a ști sigur ce studenți au participat la acel spectacol. Se are în vedere că la un spectacol nu au putut participa mai mulți elevi decât capacitatea spectacolului.

Această bază de date poate fi folosită de administratorii școlii pentru a ține evidența proceselor instituției și a introduce date noi atunci când este nevoie.

2. Testele care pot fi rulate

2.1. Testele pentru adăugări

Pentru a testa funcțiile care sunt responsabile de insert-uri în tabele, a fost făcut pachetul „TESTE_ADAUGARI” în care s-au testat funcționalitățile din pachetul „ADAUGARE”:

- **Testeaza_adaugare_sala_capacitate_invalida**: această procedură testează procedura numită adaugare.nou_sala din pachetul adaugare. Întrucât dimensiunea unei săli nu poate fi mai mică decât 10, iar capacitatea nu poate fi mai mică decât 0 sau mai mare decât 10, procedura va arunca o eroare în caz că se încearcă adăugarea unei săli cu proprietăți invalide. Astfel, procedura curentă testează dacă se adaugă o sală a cărei capacitate este -5. Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_sala_dimensiune_invalida**: această procedură testează procedura numită adaugare.nou_sala din pachetul adaugare. Întrucât dimensiunea unei săli nu poate fi mai mică decât 10, iar capacitatea nu poate fi mai mică decât 0 sau mai mare decât 10, procedura va arunca o eroare în caz că se încearcă adăugarea unei săli cu proprietăți invalide. Astfel, procedura curentă testează dacă se adaugă o sală a cărei dimensiune este 2 (mai mică decât 10). Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_spectacol_cu_capacitate_invalida**: această procedura testează procedura numită adaugare.nou_spectacol care este responsabilă de adăugarea unui spectacol nou în tabela spectacole. Întrucât capacitatea trebuie să fie un număr între 1 și 30, această procedură apelează procedura verifica_daca_capacitatea_e_corecta din pachetul verificari pentru a valida dacă spectacolul adăugat are capacitatea validă. Acest test încearcă inserarea unui spectacol cu capacitatea de -2 și se așteaptă să primească eroarea corespunzătoare. Procedura aruncă eroarea așteptată.

- **Testeaza_adaugare_elev_la_spectacol_inexistent**: această procedură este responsabilă de testarea procedurii adaugare.elev_la_spectacol care ar trebui să arunce o eroare corespunzătoare daca se introduce un spectacol care nu există. Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_elev_inexistent_la_spectacol**: această procedură este responsabilă de testarea procedurii adaugare.elev_la_spectacol care ar trebui să arunce o eroare corespunzătoare daca se introduce un elev care nu există. Procedura aruncă eroarea corespunzătoare.
- **Testeaza_adaugare_elev_nou_cu_statut_invalid**: această procedură testează procedura adaugare.nou_elev. Ea trebuie să arunce o excepție dacă statutul introdus pentru elev nu este valid. Statuturile valide sunt : „incepator", „mediu", „avansat", „profesionist". Se testează adăugarea unui elev cu statutul „lalala". Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_elev_nou_cu_varsta_invalida**: această procedură testează procedura adaugare.nou_elev. Ea trebuie să arunce o excepție dacă vârsta elevului introdus nu este corectă. Un nou elev poate avea vârsta cuprinsă între 4 și 30 de ani. Procedura încearcă să insereze un student cu vârsta de 1 an. Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_elev_nou_care_ar_trb_sa_ajunga_in_grupa_2**: această procedură testează procedura adaugare.nou_elev. Această procedură trebuie să insereze un student adăugat nou cu un anumit statut într-o grupă cu același statut unde începe. Întrucât școala de balet are doar 2 grupe de începători: 1 și 2, iar prima grupa este completă, dacă se adaugă un nou elev cu statutul de începător, acesta ar trebui introdus automat în grupa 2, fără să se specifice la inserare grupa. Acest lucru se face folosind procedura verifica_daca_incape_elev_in_grupa, deci practic se testează și această unitate. Procedura de test inserează un nou elev cu statutul de începător care ar trebui să ajungă automat în grupa 2. În urma rulării, elevul este inserat acolo unde trebuie.
- **Testeaza_adaugare_nou_instructor_pentru_grupa_care_are_deja_unul**: această procedură o testează pe cea intitulata nou_instructor. Pentru a se adăuga un nou instructor pentru o grupă, acea grupă nu trebuie să aibă deja un instructor asignat. Dacă are, la încercarea inserării, trebuie primită o eroare corespunzătoare. Procedura arunca eroarea așteptată.
- **Testeaza_adaugare_nou_instructor**: testează procedura nou_instructor. Aceasta mai întâi șterge instructorul corespunzător grupei 8 pentru a se asigura ca acea grupă nu are instructor și apoi inserează unul pentru ea. Procedura are rezultatele așteptate.
- **Testeaza_adaugare_nou_instructor_care_nu_corespunde_grupei**: la inserarea unui instructor pentru o anumită grupă, atât grupa cât și instructorul au câte un status. Pentru a putea asigna un instructor grupei, acestea trebuie să aibă același statut. Procedura nou_instructor se asigură și de acest lucru. Această procedură aruncă eroare atunci când se încearcă inserarea unui instructor unei grupe a cărei statut nu se potrivește. Procedura aruncă eroarea așteptată dacă se dorește asignarea unui instructor începător unei grupe de profesioniști.
- Se testează aceleași funcționalități ca mai sus, dar pentru procedura nou_pianist, dar a cărei condiții sunt aceleași ca mai sus prin procedurile:
  Testeaza_adaugare_nou_pianist_pentru_grupa_care_are_deja_unul, Testeaza_adaugare_nou_ pianist, Testeaza_adaugare_nou_ pianist _care_nu_corespunde_grupei
- **Testeaza_adaugare_noua_grupa_statut_invalid:** procedura testează procedura noua_grupa care ar trebui să arunce o eroare dacă se introduce o grupă care are un statut diferit de „începător",„mediu",„avansat",„profesionist". Se introduce o grupă cu statut „lalala", iar procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_noua_grupa_in_sala_inexistenta:** înainte de adăugarea unei grupe noi, se verifică dacă sala în care se încearcă adăugarea grupei există. Se folosește astfel procedura din pachetul verificari.verifica_daca_exista_sala. Procedura aruncă eroarea așteptată dacă se asignează o sală care nu există (9999 în cazul de față).
- **Testeaza_adaugare_noua_grupa:** procedura care testează dacă atunci când se adaugă o grupă validă, totul este în regulă. Se introduce astfel sala cu statut de „profesionist" în sala 3. Inserarea are loc cu succes.

- **Testeaza_adaugare_sala:** procedura care testează introducerea unei săli cu succes cu funcția adaugare.nou_sala. Mai întâi se șterge dacă există sala cu dimensiunea 66 și capacitatea 6, apoi se apelează procedura pentru a se adăuga noua sală. Operația are succes.

2.2. Testele pentru actualizări

Pentru a testa funcțiile care sunt responsabile de update-uri în tabele, a fost făcut pachetul „TESTE_ACTUALIZARI” în care s-au testat funcționalitățile din pachetul „ACTUALIZARI”:

- **Testeaza_modifica_experienta_instructor:** este o procedură care o testează pe cea din pachetul actualizari intitulată modifica_experienta_instructor care va modifica experiența instructorului cu id-ul 7 de la 20 la 21. Noua experiență trebuie să fie mai mare decât cea veche, dar pentru această verificare este responsabil trigger-ul numit actualizare_experienta_instructori care va fi testata în pachetul pentru triggeri. Procedura are succes.
- **Testeaza_modifica_experienta_pianist:** este o procedură care o testează pe cea din pachetul actualizari intitulată modifica_experienta_pianist care va modifica experiența pianistului cu id-ul 7 de la 20 la 21. Noua experiență trebuie să fie mai mare decât cea veche, dar pentru această verificare este responsabil trigger-ul numit actualizare_experienta_pianisti care va fi testata în pachetul pentru triggeri. Procedura are succes.
- **Testeaza_modifica_statut_grupa**: este o procedura care o testează pe cea din pachetul actualizari numită modifica_statut_grupa. Face acest lucru prin selectarea grupei cu cei mai mare id, mai întâi o actualizează la statutul începător pentru a se asigura ca are acest statut. Apoi, introduce un nou elev cu numele „ccc” și prenumele „ccc” cu statutul „începător” în grupa modificată mai devreme. Apoi, modifică iar statutul grupei în „profesionist”. Se observă că se modifică atât statutul grupei, cât și cel al studenților care fac parte din respectiva grupă. Testul are succes.
- **Testeaza_modifica_statut_invalid_grupa**: atunci când se actualizează statutul unei grupe, trebuie ca metoda responsabilă de acest lucru să se asigure că statutul este valid. Această procedură demonstrează ca nu se poate schimba statutul cu unul invalid cum ar fi „lalala”. Se primește eroarea așteptată.

2.3. Testele pentru ștergeri

Pentru a testa funcțiile care sunt responsabile de ștergerile din tabele, a fost făcut pachetul „TESTE_STERGERI” în care s-au testat funcționalitățile din pachetul „STERGERI”:

- **Testeaza_stergere_elev_inexistent:** testează procedura ștergeri.elev care este responsabilă cu ștergerea unui elev. Dacă acest elev nu există, trebuie aruncată o eroare. Testarea constă în încercarea de a șterge un elev cu id-ul 9999 , iar în urma acestuia se primește eroarea așteptată.
- **Testeaza_stergere_elev_care_are_spectacole:** aceasta este responsabilă cu testarea ștergerii unui elev care are spectacole în tabela elevi_spectacol. Procedura care face ștergerea este responsabilă și de ștergerea din tabela elevi_spectacol, de adăugarea intrărilor șterse în tabela de istoric și apoi de ștergerea elevului din tabela elevi. Astfel, se inserează un elev nou cu numele „test” și prenumele „test” cu statutul „profesionist” în grupa 8 care este de profesioniști. Apoi, se adaugă acest elev la spectacolul „Nopți albe”. Apoi, se șterge elevul. Rezultatele procedurii sunt cele așteptate: elevul este șters din tabela elevi, sunt șterse și spectacolele lui din tabela elevi_spectacol și acestea sunt introduse în tabela de istoric.
- **Testeaza_stergere_elev_spectacole:** se presupune că ștergerile din această tabelă au loc daca din greșeală s-a introdus un elev care nu a participat cu adevărat la acel spectacol. Deci, doar se va insera un nou elev (10) la spectacolul Nopți albe și apoi se va șterge. Procedura de testare se execută cu succes.

- **Testeaza_ștergere_spectacol:** se presupune în aceeași manieră că ștergerea unui spectacol din această tabelă are loc după ce inserarea s-a întâmplat din greșeală având în vedere tabela conține spectacole care au avut deja loc. Deci, se inserează spectacolul cu numele NouSpect și apoi se șterge. Operațiile au loc cu succes.
- **Testeaza_ștergere_spectacol_inexistent:** se presupune că această procedură aruncă o eroare dacă se încearcă ștergerea unui spectacol care nu există. Deci se încearcă ștergerea spectacolului denumit NUEXISTA, iar testul returnează eroarea așteptată.
- Procedurile **testeaza_ștergere_instructor_inexistent** și **testeaza_ștergere_pianist_inexistent** fac același lucru doar că încearcă să șteargă instructorul sau pianistul cu id-ul 9999 , testele au rezultatele așteptate.
- **Testeaza_stergere_instructor:** procedură care o testează pe cea intitulată stergeri.instructor care mai întâi apelează procedura adaugare.nou_instructor(...) și apoi șterge instructorul de abia adăugat. Procedura se execută cu succes.
- **Testeaza_stergere_pianist:** procedură care o testează pe cea intitulată stergeri.pianist care mai întâi apelează procedura adaugare.nou_pianist(...) și apoi șterge pianistul de abia adăugat. Procedura se execută cu succes.

2.4. Testele pentru triggeri

Pentru a testa funcțiile care sunt responsabile de ștergerile din tabele, a fost făcut pachetul „TESTE_TRIGGERI” în care s-au testat funcționalitățile triggerilor:

- **Testeaza_trigger_actualizare_experienta_cadru_didactic_crapa**: procedură care testează triggerul actualizare_experienta_instructori. Astfel, se încearcă modificarea experienței unui instructor cu un număr de ani mai mic decât avea înainte. Acest lucru este imposibil, așa că triggerul este responsabil de a avertiza când se întâmplă asta și de a nu permite operația. Procedura curentă încearcă să modifice experiența instructorului cu id-ul 5 la 6, deși experiența lui curentă este de 20. Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_spectacol_data_in_viitor:** procedură care testează triggerul trg_data_spectacol care nu lasă administratorii să insereze un spectacol care nu a avut loc încă. Astfel se încearcă introducerea unui spectacol în anul 2024. Procedura aruncă eroarea așteptată.
- **Testeaza_introducere_elev_nou_grupa_unde_nu_mai_are_loc:** procedură care testează triggerul numit verifica_daca_se_poate_introduce_elev_nou_in_grupa. Acest trigger verifică dacă mai este loc în grupa respectivă (folosindu-se de procedura verifica_daca_incape_elev_in_grupa(...)) care practic verifică dacă capacitatea sălii în care este asignată grupa nu este depășită. Procedura aruncă eroarea așteptată.
- **Testeaza_adaugare_elev_in_spectacol_capacitate_atinsa_deja:** se testează adăugarea unui participant la un spectacol unde deja s-au introdus toți elevii care au participat la acel spectacol (fie că în tabela de istoric, fie în cea de elevi_spectacol). Procedura șterge mai întâi intrările din elevi_spectacol și elevi_spectacol_istoric intrările corespunzătoare spectacolul denumit Ghioceii. Acesta, având capacitatea de 3, inserează 2 spectacole în elevi_spectacol și unul în elevi_spectacol_istoric pentru a arăta că se iau în considerare și cele din această tabelă. Apoi, mai inserează un al patrulea elev pentru acest spectacol și obține eroarea așteptată provenită din trigger, iar elevul nu mai este adăugat pentru acest spectacol. Nu se ia în considerare și adăugarea în tabela elevi_spectacol_istoric în cadrul triggerului deoarece se presupune că nu va fi introdus în istoric un spectacol care nu a fost deja în tabela elevi_spectacol.

2.5 Testele pentru vizualizări

- testeaza_afisare_elevi
- testeaza_afisare_elevi_din_grupa_5
- testeaza_afisare_elevi_din_grupa_invalida

- testeaza_afisare_grupe_din_sala_invalida
- testeaza_afisare_pianistul_grupei_1
- testeaza_afisare_pianist_grupa_invalida
- testeaza_afisare_spectacole_la_care_nu_au_fost_introdusi_elevi
- testeaza_afisare_grupe_in_care_mai_incap_elevi

2.6 Testele pentru tranzacție

Tranzacția proiectului constă în avansarea unui elev. Practic, când se hotărăște ca un student să aibă un statut mai avansat, trebuie să i se modifice statutul dar și grupa în care este. La avansarea unui elev se pot lua în considerare și numărul de spectacole la care a participat după ultima avansare, așa că, în tabela de elevi_spectacol odată cu avansarea se vor șterge înregistrările corespondente studentului în cauză și se vor adăuga în tabela de istoric. I se va modifica statutul (folosind o funcție care returnează care este statutul următor) și apoi va fi mutat într-o grupă în care are loc cu același statut ca noul său nivel. Dacă este deja profesionist, nu se schimbă nimic. Dacă nu mai are loc în nicio grupă care să aibă nivelul căutat, nu va fi produsă nicio schimbare până când se va hotărî reîncercarea modificării.

Tranzacția se află în pachetul avansare_elev, iar procedura în care are loc se numește modifica. Se apelează procedura modifica din pachetul avansare_elev. Acesta conține o funcție privată care în funcție de statutul curent al elevului, returnează următorul nivel. De exemplu, dacă studentul este începător, următoarea treaptă ar fi mediu. Această procedură:

- Șterge cele spectacolele elevului din elevi_spectacol și le adaugă la tabela de istoric
- Schimbă statutul elevului
- În funcție de statutul nou al elevului i se caută o nouă grupă cu statutul corespunzător. Se extrag grupele care au statutul căutat și apoi folosindu-se de procedura verifica_daca_incape_elev_in_grupa se alege în ce grupă se va muta elevul.
- Se mută elevul în grupa respectivă
- Dacă nu s-a găsit nicio grupă în care are loc, va rămâne la același statut.
- Dacă elevul este deja profesionist, nu va mai avansa, iar spectacolele lui nu vor fi adăugate în istoric.

Testele disponibile sunt:

- **Testeaza_tranzactie_avansare_elev_reusita_cu_succes**
- **Testeaza_tranzactie_cu_elev_deja_profesionist :** dacă elevul este deja profesionist, nu ar trebui să se producă nicio schimbare. Testul are rezultatul așteptat.
- **Testeaza_avansare_elev_care_nu_mai_are_loc_in_grupa_avansata**: presupune că avem un elev cu statut mediu și vrem să îl avansăm în grupa de avansați. Totuși, în nicio grupă cu acest statut nu mai este loc, așa că niciuna dintre operațiile tranzacției nu se va executa.

1. Structura și inter-relaționarea tabelelor

1.1. Descrierea structurii

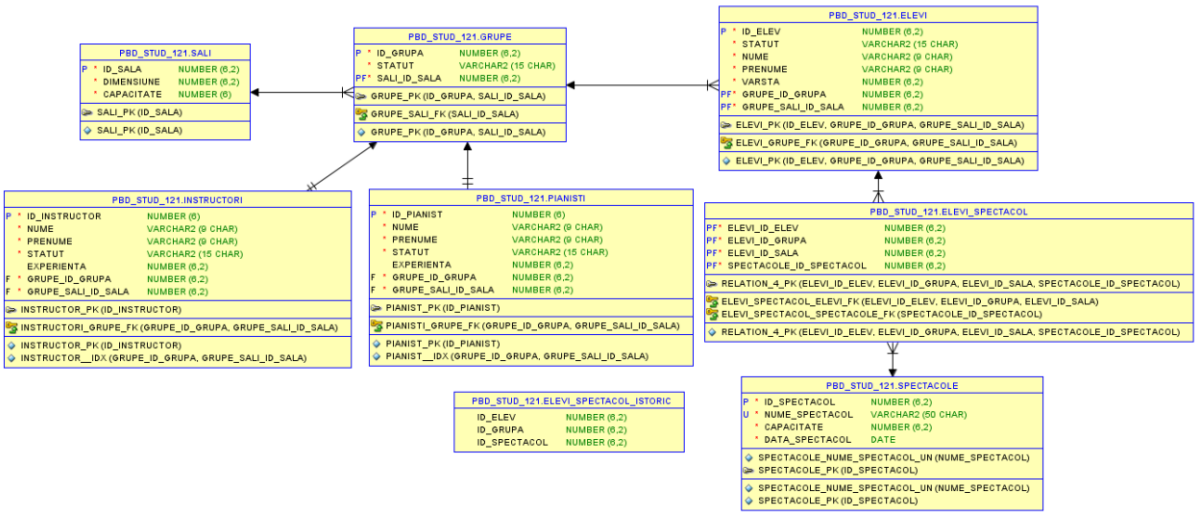
Tabelele din aceasta aplicatie sunt:

- Sali
- Grupe
- Instructori
- Pianisti
- Elevi
- Spectacole

- ‘elevi_spectacol’
- Elevi_spectacol_istoric

Intre tabelele Sali si Grupe, exista o relatie one to many (1:n). La fel si intre Grupe si Elevi. Avem si doua relatii one-to-one intre Grupe si Instructori si Grupe si Pianisti. Relatia many to many poate fi observata intre tabelele Elevi si Spectacole. Între tabelul elevi_spectacol_istoric nu este nicio relație deoarece se ia în considerare cazul în care se dorește ștergerea unui elev, a unei grupe sau a unui spectacol (inclusiv vrem să vedem în ce grupă era elevul când a participat la respectivul spectacol, nu în ce grupă este acum pentru eventuale statistici).

1.2. Inter-relaționarea tabelelor



Observam ca relatia de many to many dintre tabelele Elevi si Spectacole a fost sparta in doua relatii one to many (1:n) , iar intre cele doua tabele a fost introdus un nou tabel elevi_spectacol care monitorizeaza la ce spectacole a participat un anumit elev. Sunt 2 relatii de 1:1 intre Grupe:Instructori si Grupe:Pianisti, restul relatiilor fiind de 1:n.

Fiecare id din fiecare tabela este introdus cu ajutorul autoincrementului : id_sala,id_grupa, id_elev, id_instructor, id_pianist, id_spectacol. Acest lucru ne ajuta sa nu mai introducem manual id-ul, asigurandu-ne ca fiecare id va avea o valoare diferita.

Relațiile dintre tabele sunt caracterizate prin foreign keys care în cazul de față sunt:

• In tabela Grupe

- O constrangere de tip Foreign Key pentru Sali_id_sala care arata ca Sali_id_sala este extras din tabela Sali (fiind o relatie de 1:n intre Grupe:Sali) unde id_sala este de tip primary

• In tabela Instructori:

- O constrangere de tip foreign key pentru Grupe_id_grupa care se datoreazarelatiei de 1:1 intre instructori:grupe, id-ul grupei fiind preluat astfel in aceastacoloana

- Inca o constrangere de tip foreign key pentru Grupe_Sali_id_sala care se datoreaza faptului ca in tabela grupe cu care are relatie de 1:1 este un primary (foreign) key pe Sali_id_sala

• In tabela Pianisti:

- O constrangere de tip foreign key pentru Grupe_id_grupa care se datoreaza relatiei de 1:1 intre pianisti:grupe, id-ul grupei fiind preluat astfel in aceasta coloana

- Inca o constrangere de tip foreign key pentru Grupe_Sali_id_sala care se datoreaza faptului ca in tabela grupe cu care are relatie de 1:1 este un primary (foreign) key pe Sali_id_sala

• In tabela Elevi:

- O constrangere de tip foreign key (care este si primary) pentru coloana Grupe_id_grupa care se datoreaza legaturii de 1:n dintre Elevi:Grupe, fiind preluat id-ul grupei pentru a putea stii in ce grupa se afla elevul

- O constrangere de tip foreign key (care este si primary) pentru coloana Grupe_Sali_id_sala care se datoreaza relatiei dintre elevi si grupe , grupe avand ca primary key(foreign) care provine din tabela Sali atributul Sali_id_sala care a fost preluat de coloana noastra ca Grupe_Sali_id_sala

• In tabela elevi_spectacol:

- Avem 4 constrangeri de tip Primary Foreign Key care sunt preluate din tabelele Elevi si Spectacole deoarece aceasta tabela este cea care face legatura dintre Elevi si Spectacole. Ea a rezultat in urma spargerii legaturii dem:n, fiind astfel in legaturi de 1:n cu 1:Elevi si 1:n cu 1:Spectacole, avand ca coloane: Elevi_id_elev, Elevi_id_grupa, Elevi_id_sala, Spectacole_id_spectacol, toate fiind PF-uri.

2. Descrierea logicii stocate

2.1. Pachete, funcții, proceduri

Proceduri

- **PROCEDURA verifica_daca_nu_a_fost_atinsa_capacitatea:** verifică dacă pentru un anumit spectacol a fost atinsă sau nu capacitatea numărând atât elevii din elevi_spectacol care au participat la respectivul spectacol, cât și pe cei din elevi_spectacol_istoric. Primește ca parametrul id-ul spectacolului ca parametru de tip IN și unul de tip OUT pentru a prelua flag-ul care spune dacă a fost atinsă capacitatea sau nu.
  - **RECORDs:**
    - **typ_spect_rec:** conține date de tipul capacității din spectacole
    - **Typ_nr_rec:** conține date de tipul NUMERIC
  - **CURSORI:**
    - (restrictiv) **typ_spect** de tipul typ_spect_rec folosit pentru capac care va prelua capacitatea spectacolului
    - (restrictiv) **typ_nr** de tipul typ_nr_rec folosit pentru variabila nr_elevi care va prelua numărul de elevi care au participat la spectacol
  - **EXCEPȚII:**
    - **excSpectacol:** excepție aruncată dacă nu este găsit spectacolul căutat
- **PROCEDURA verifica_daca_incape_elev_in_grupa:** verifică dacă într-o anumită grupă mai este loc pentru un elev luând în considerare capacitatea sălii în care este asignată grupa. Are un parametru de tip OUT BOOLEAN pentru flag-ul corespondent daca se poate adăuga un nou elev și unul de tip IN pentru id-ul grupei.

Funcții

- **FUNCȚIA numara_la_cate_spectacole_a_participat_un_elev:** primește ca parametru id-ul elevului și returnează numărul de spectacole la care a participat (din elevi_spectacol doar deoarece funcția aceasta ne va folosi în trigger-ul care va fi responsabil de actualizarea statutului elevilor)

Pachete

- **PACHETUL Verificări** care conține:

- **PROCEDURA Verifica_daca_exista_elev:** procedură care are rolul de a arunca excepție dacă nu există elevul căutat primit ca parametru de intrare(id-ul).
    - **CURSORI:**
        - **C:** cursor static care preia numele elevului cu un anumit id
    - **EXCEPȚII:**
        - **Inexistent:** excepția aruncată dacă nu există elevul căutat, -20003
- **PROCEDURA Verifica_daca_exista_deja_spectacol_cu_acest_nume:** procedură care are rolul de a arunca excepție dacă nu există spectacolul cu numele căutat primit ca parametru IN
    - **EXCEPȚII:**
        - **Spectacol_nu_exista:** excepția aruncată dacă nu există spectacolul căutat, -20013

- **PROCEDURA Verifica_daca_capacitatea_e_corecta:** verifică dacă capacitatea spectacolului primită ca parametru de intrare este între 1 și 30
    - **EXCEPȚII:**
        - **Capacitate_invalida:** excepția aruncată dacă nu este validă, -20070
- **PROCEDURA Verifica_daca_exista_sala:** procedură care primește id-ul sălii ca parametru de intrare și aruncă o excepție dacă nu a fost găsită
    - **EXCEPȚII:**
        - **Nu_exista_sala:** excepția aruncată, -20058
- **PROCEDURA Verifica_daca_exista_instructor:** procedură care primește id-ul instructorului ca parametru de intrare și aruncă o excepție dacă nu a fost găsit instructorul
    - **EXCEPȚII:**
        - **Nu_exista_instructor:** excepția aruncată, -20050
- **PROCEDURA Verifica_daca_exista_pianist:** procedură care primește id-ul pianistului ca parametru de intrare și aruncă o excepție dacă nu a fost găsit
    - **EXCEPȚII:**
        - **Nu_exista_pianist:** excepția aruncată, -20051
- **PROCEDURA Verifica_daca_exista_grupa:** procedură care primește id-ul grupei ca parametru de intrare și aruncă o excepție dacă nu a fost găsită grupa
    - **EXCEPȚII:**
        - **Nu_exista_grupa:** excepția aruncată dacă nu există grupa, -20049
- **PROCEDURA Verifica_daca_grupa_are_nevoie_de_instructor:** procedură care primește id-ul grupei și returnează o excepție dacă grupa căutată are deja un instructor asignat
    - **EXCEPȚII:**
        - **Nu_are_nevoie:** excepția aruncată dacă postul nu este liber, -20036
- **PROCEDURA Verifica_daca_grupa_are_nevoie_de_pianist:** procedură care primește id-ul grupei și returnează o excepție dacă grupa căutată are deja un pianist asignat (postul nu este liber)
    - **EXCEPȚII:**
        - **Nu_are_nevoie:** excepția aruncată dacă postul nu este liber, -20035
- **PROCEDURA Verifica_daca_statutul_corespunde_grupei:** procedură ce primește ca parametrii de intrare id-ul grupei și statutul căutat și aruncă o excepție dacă acesta e diferit de statutul grupei.
    - **EXCEPȚII:**
        - **Nu_corespunde**: excepția aruncată dacă nu corespunde, -20069

❖ **PACHETUL avansare_elevi** care va conține:

- **FUNCȚIA PRIVATĂ urmatorul_statut**: care primește ca parametru statutul curent și îl returnează pe următorul.
    - **EXCEPȚII:**
        - **Statut inexistent:** excepție aruncată dacă statutul nu există, -20006

        - **Nu_mai_poate_avansa:** excepție aruncată dacă elevul este deja profesionist, -20010
    - **PROCEDURA PUBLICĂ modifica:** Șterge cele 5 spectacole ale elevului din elevi_spectacol și le adaugă la tabela de istoric. Schimbă statutul elevului. În funcție de statutul nou al elevului i se caută o nouă grupă cu statutul corespunzător. Se extrag grupele care au statutul căutat și apoi folosindu-se de procedura verifica_daca_incape_elev_in_grupa se alege în ce grupă se va muta elevul. Se mută elevul în grupa respectivă. Dacă nu s-a găsit nicio grupă în care are loc, va rămâne la același statut până următoarea dată când va fi participat la încă 5 spectacole.
        - **RECORDs:**
            - **Typ_spect_rec:** compus din id_grp care are tipul de date al id-ului grupei
        - **TABLEs:**
            - **Data_table_type:** compus din date de tipul id-ului grupei și cu index de tip BINARY INTEGER
        - **CURSORI:**
            - **(restrictiv) Typ_spect:** de tipul typ_spect_rec, variabila grp care este responsabilă pentru selectarea grupelor care au statutul corespondent cu noul statut al elevului.
            - **(static) c_spect:** responsabil de stocarea intrărilor din elevi_spectacol la care au participat elevul în cauză
        - **EXCEPȚII:**
            - **Inca_nu_avanseaza:** excepție aruncată dacă nu s-a găsit grupă disponibilă (care să mai aibă locuri) cu statutul căutat
            - **Nu_exista_grupe_cu_acest_statut:** excepție aruncată dacă nu există grupe care au statutul căutat

❖ **PACHETUL adaugare** care conține:

- **PARAMETRU PRIVAT:** excepția statut_invalid (deoarece va fi folosită de mai multe proceduri), -20082
- **PROCEDURA elev_la_spectacol:** procedură responsabilă de inserarea unui elev la un spectacol. Primește ca parametrii de intrare id-ul elevului și numele spectacolului la care acesta a participat.
    - **RECORDs:**
        - **Typ_spect_rec:** conține tip de dată de tipul id-ului spectacolului
        - **Typ_grup_rec:** conține tip de dată de tipul id-ului grupei și al id-ului sălii
    - **CURSORI:**
        - **(restrictiv) Typ_spect:** în variabila spect care va reține spectacolul cu numele din parametrii de intrare
        - **(restrictiv) Typ_grup:** în variabila grup care va reține id-ul grupei și id-ul sălii în care este repartizată aceasta corespunzătoare elevului căutat
    - **EXCEPȚII:**
        - **Nu_exista_spectacol:** dacă nu există spectacolul cu numele dat, -20001
        - **Nu_exista_elev:** dacă nu există elevul pe care dorim să îl introducem, -20003
- **PROCEDURA nou_spectacol:** procedură responsabilă de inserarea unui nou spectacol. Primește ca parametrii de intrare numele, capacitatea și data pe care va avea loc. Se vor apela pentru verifică procedurile verifica_daca_exista_deja_spectacol_cu_acest_nume și verifica_daca_capacitatea_este_corecta.
- **PROCEDURA nou_elev:** procedură responsabilă de inserarea unui nou elev. Primește ca parametrii de intrare statutul, numele, prenumele și vârsta elevului. Nu este specificată grupa deoarece procedura, în funcție de statutul elevului, va găsi o grupă cu același statut și cu locuri disponibile în care să îl insereze (folosind procedura

verifica_daca_incape_elev_in_grupa). Dacă nu se găsește o grupă, nu se va adăuga elevul și se va arunca o eroare.
    - **CURSORI:**
        - **C:** cursor care va stoca grupele cu statutul căutat
    - **EXCEPȚII:**
        - **Varsta_invalida:** dacă vârsta introdusă este mai mică decât 4 sau mai mare decât 30 atunci este aruncată această eroare, -20081
        - **Statut_invalid:** excepția aruncată dacă statutul este invalid (param privat)
- **PROCEDURA nou_instructor:** procedură responsabilă de inserarea unui nou instructor. Primește ca parametrii de intrare numele, prenumele, statutul, experiența și grupa noului instructor. Procedura aceasta apelează din pachetul verificări procedurile verifica_daca_exista_grupa, verifica_daca_statutul_corespunde_grupei, verifica_daca_grupa_are_nevoie_de_instructor.
    - **EXCEPȚII:**
        - **Experienta_invalida:** dacă experiența are valoare negativă, atunci este invalidă și se aruncă această excepție, -20076
        - **Statut_invalid:** excepția aruncată dacă statutul este invalid (param privat)
- **PROCEDURA nou_pianist:** procedură responsabilă de inserarea unui nou pianist. Primește ca parametrii de intrare numele, prenumele, statutul, experiența și grupa noului pianist. Procedura aceasta apelează din pachetul verificări procedurile verifica_daca_exista_grupa, verifica_daca_statutul_corespunde_grupei, verifica_daca_grupa_are_nevoie_de_pianist.
    - **EXCEPȚII:**
        - **Experienta_invalida:** dacă experiența are valoare negativă, atunci este invalidă și se aruncă această excepție, -20076
        - **Statut_invalid:** excepția aruncată dacă statutul este invalid (param privat)
- **PROCEDURA nou_grupa:** procedură responsabilă de inserarea unei noi grupe. Primește ca parametrii de intrare statutul și sala căreia îi va fi asignată grupa.
    - **EXCEPȚII:**
        - **Statut_invalid:** excepția care e variabila privată a pachetului, este aruncată dacă statutul introdus pentru noua grupă nu este valid
- **PROCEDURA nou_sala:** procedură responsabilă de inserarea unei noi săli. Primește ca parametrii de intrare dimensiunea și capacitatea sălii.
    - **EXCEPȚII:**
        - **Dimensiune_invalida:** dacă dimensiunea primită ca parametru este mai mică decât 10, este invalidă și este aruncată excepția -20057
        - **Capacitate_invalida:** dacă capacitatea primită ca parametru este mai mică decât 0 sau mai mare ca 10, atunci este invalidă și este aruncată excepția -20052
- **PROCEDURA elev_spect_istoric:** procedura care inserează date în tabela de istoric. Primește ca parametrii id-ul elevului, id-ul grupei și id-ul spectacolului

❖ **PACHETUL teste_adaugari (descris în primul capitol)**

❖ **PACHETUL actualizari** care conține:

- Variabilă privată excepție **statut_invalid**, -20082
- **PROCEDURA modifica_experienta_pianist:** responsabilă de modificarea experienței pianistului. Primește ca parametrii de intrare id-ul pianistului și noua experiență. Folosește metoda din pachetul verificări care verifica_daca_exista_pianist și modifică experiența.
- **PROCEDURA modifica_experienta_instructor:** responsabilă de modificarea experienței instructorului. Primește ca parametrii de intrare id-ul instructorului și noua experiență. Folosește metoda din pachetul verificări care verifica_daca_exista_instructor și modifică experiența.

- **PROCEDURA modifica_statut_grupa:** responsabilă de modificarea statutului unei grupe și odată cu această actualizare, cu modificarea statutului tuturor elevilor din acea grupă. Primește ca parametrii de intrare grupa și noul statut.
    - **CURSORI:**
        - **(static) C:** va conține id-urile elevilor din grupa a cărui statut se modifică
    - **EXCEPȚII:**
        - **Statut_invalid:** excepția care e variabila privată a pachetului, este aruncată dacă statutul introdus pentru noua grupă nu este valid

❖ **PACHETUL teste_actualizări (descris în primul capitol)**

❖ **PACHETUL stergeri** care conține:

- **PROCEDURA instructor:** responsabilă de ștergerea unui instructor. Primește ca parametrii de intrare id-ul instructorului. Folosește procedura verifica_daca_exista_instructor.
- **PROCEDURA pianist:** responsabilă de ștergerea unui pianist. Primește ca parametrii de intrare id-ul pianistului. Folosește procedura verifica_daca_exista_pianist.
- **PROCEDURA spectacol:** responsabilă de ștergerea unui spectacol. Primește ca parametrii de intrare numele spectacolului. Folosește procedura verifica_daca_exista_spectacol.
- **PROCEDURA elevi_spectacol:** responsabilă de ștergerea unei intrări din elev_spectacol corespunzătoare participării unui elev la un anumit spectacol. Primește ca parametrii de intrare id-ul elevului și numele spectacolului. Folosește procedura verifica_daca_exista_spectacol_cu_acest_nume și verifica_daca_exista_elev.
- **PROCEDURA elev:** responsabilă de ștergerea unui elev. Primește ca parametrii id-ul elevului. Mai întâi apelează funcția verifica_daca_exista_elev. Apoi, ia toate spectacolele din elevi_spectacol cu ajutorul cursorului și le introduce în tabela de istoric. Pe urmă, șterge elevul.
    - **CURSORI:**
        - **(static) c:** responsabil cu stocarea intrărilor din elevi_spectacol care îl conțin pe elevul în cauză

❖ **PACHETUL teste_ștergeri (descris în capitolul 1)**

❖ **PACHETUL testare_triggeri (descris în capitolul 1)**

❖ **PACHETUL vizualizari** care conține:

- **PROCEDURA afiseaza_grupe_in_care_mai_incap_elevi:** responsabilă de afișarea grupelor care mai au locuri disponibile pentru elevi
    - **CURSORI:**
        - **(static) c:** returnează toate grupele
- **PROCEDURA afiseaza_spectacolele_la_care_nu_au_fost_introdusi_elevi:** procedură responsabilă de afișarea spectacolelor la care nu a fost introdus niciun elev participant
    - **CURSORI:**
        - **(static) c:** returnează toate spectacolele
- **PROCEDURA afiseaza_pianistul_unei_grupe:** procedura care afișează care este pianistul unei grupe dată ca parametru de intrare prin id-ul său.
    - **CURSORI:**
        - **(static) c:** returnează pianistul care corespunde grupei
- **PROCEDURA afiseaza_elevi:** procedura care afișează toți elevii din școala de balet.
    - **CURSORI:**
        - **(static) c:** returnează toți elevii existenți
- **PROCEDURA afiseaza_elevi_din_anumita_grupa:** procedură care afișează elevii dintr-o anumită grupă dată ca parametru de intrare prin id-ul grupei
    - **CURSORI:**
        - **(static) c:** returnează elevii unei anumite grupe
- **PROCEDURA afiseaza_grupele_asignate_unei_anumite_Sali:** procedură care afișează grupele care sunt asignate unei anumite săli primită ca parametru de intrare

2.2. Triggeri

- **Trg_data_spectacol** : trigger care este responsabil pentru asigurarea că în spectacole nu se introduc intrări cu date calendaristice în viitor (se pot introduce spectacole care au avut loc deja). Acesta este apelat înainte de insert sau update pe tabela spectacole. Dacă data nu corespunde condiției, se aruncă excepția cu numărul -20001 cu mesajul „Data invalida ... trebuie să fie mai mare decât data curentă”)
- **Verifica_daca_se_poate_introduce_elev_nou_in_grupa:** trigger apelat înainte de actualizare sau inserare în tabela elevi pe câmpul corespondent id-ului grupei care este responsabil de asigurarea că capacitatea grupei nu este atinsă
- **Adaugare_elev_in_spectacol:** trigger care este apelat înainte de introducerea datelor în elevi_spectacol care verifică dacă capacitatea spectacolului nu a fost deja atinsă folosind procedura verifica_daca_nu_a_fost_atinsa_capacitatea. Dacă a fost deja atinsă, se va arunca excepția -20001, iar elevul nu va mai fi adăugat la spectacol.
- **Actualizare_experienta_instructori:** trigger care verifică ca înainte să fie făcut un update pe tabela instructori pe câmpul de experiență, să fie verificat ca noua experiență introdusă să nu aibă valoare mai mică decât cea veche. (nu se poate ca experiența să scadă). Dacă este mai mică, nu se va efectua actualizarea și se va arunca excepția **experienta_exceptie** -20033.
- **Actualizare_experienta_pianist:** trigger care verifică ca înainte să fie făcut un update pe tabela pianiști pe câmpul de experiență, să fie verificat ca noua experiență introdusă să nu aibă valoare mai mică decât cea veche. (nu se poate ca experiența să scadă). Dacă este mai mică, nu se va efectua actualizarea și se va arunca excepția **experienta_exceptie** -20033.