Gestionarea bazei de date a unei biblioteci

Sabău Eduard, Grupa 134

1. Modelul real, utilitatea acestuia și reguli de funcționare.

Biblioteca este o instituție care îi permite unui membru să împrumute cărți pentru o perioadă specifică de timp, întârzierea fiind penalizată, să rezerve cărți începând cu data în care sunt returnate dacă nu sunt disponibile în acel moment pentru împrumut, să utilizeze cărțile în sala de lectură din incintă și alte servicii precum xerox, contra-cost. Aceasta funcționează pe bază de abonament pe perioade de timp.

Utilitatea unei biblioteci constă în faptul că oferă accesul la educație și persoanelor ce nu dispun de capitalul necesar pentru a achiziționa cărți noi, unele categorii de persoane având acces gratuit în foarte multe instituții de acest fel. De asemenea, biblioteca reprezintă un loc prietenos pentru cei ce doresc să studieze în liniște întrucât le sunt puse la dispoziție sălile de lectură.

Reguli de funcționare:

Fiecare carte are un ID unic și poate avea mai mulți autori.

Biblioteca poate avea mai multe copii ale aceleiași cărți.

Membrii pot alege între tipuri de abonament ce au perioade diferite de valabilitate și pot vedea istoricul abonamentelor.

Membrii pot vedea data aderării și când au împrumutat o anumită carte.

Membrii pot primi penalizări dacă nu returnează cartea în 14 zile de la data împrumutului, penalizări pe care le pot plăti.

Membrii pot căuta cărțile după titlu, autori, ISBN, gen, limbă sau data în apariției.

Membrii pot împrumuta cărți, verifica dacă o carte poate fi împrumutată și pot rezerva cărți dacă toate copiile au fost deja împrumutate.

Personalul poate vedea istoricul abonamentelor, împrumuturilor și penalizărilor tuturor membrilor.

1. Constrângerile impuse asupra modelului

O carte trebuie să aibă minim un autor, dar poate avea și mai mulți.

O carte trebuie să aparțină minim unui gen, dar poate aparține mai multor genuri.

O carte poate fi împrumutată de mai mulți membri (nu în același timp).

O carte poate fi rezervată de mai mulți membri (nu în același timp).

Unui gen îi pot aparține mai multe cărți.

Un autor poate scrie mai multe cărți.

Un membru poate împrumuta mai multe cărți.

Un membru poate rezerva mai multe cărți.

Un membru poate primi mai multe penalizări.

Un membru trebuie să cumpere minim un abonament, căruia obligatoriu îi trebuie asociată o plată, dar poate cumpăra mai multe.

O penalizare poate fi atribuită unui singur membru.

O penalizare poate fi plătită o singură dată.

O plată de penalizare trebuie să fie atribuită unei singure penalizări.

Un abonament poate fi cumpărat de mai mulți membri și atribuit mai multor plăți.

O plată de abonament poate fi atribuită mai multor abonamente și mai multor membri.

1. Descrierea entităților
   1. Entitatea CARTE

Entitatea este independentă și reprezintă orice carte din inventarul bibliotecii care poate fi împrumutată sau rezervată. Cheia primară a entității este *cod\_carte*.

* 1. Entitatea GEN

Entitatea este independentă și reprezintă genul căruia îi poate aparține o carte. Cheia primară a entității este *cod\_gen*.

* 1. Entitatea AUTOR

Entitatea este independentă și reprezintă orice autor al unei cărți prezente sau nu în inventarul curent al bibliotecii. Cheia primară a entității este *cod\_autor*.

* 1. Entitatea MEMBRU

Entitatea este independentă și reprezintă orice membru ce a cumpărat minim un abonament și este eligibil pentru a împrumuta și rezerva cărți. Cheia primară a entității este *cod\_membru*.

* 1. Entitatea PENALIZARE

Entitatea este dependentă de entitatea MEMBRU întrucât cheia sa primară este compusă din perechea *cod\_penalizare* și *cod\_membru* (cheia primară a entității MEMBRU) și reprezintă penalizările pe care le pot primi membri pentru returnarea peste termen a cărților sau deteriorarea acestora.

* 1. Entitatea PLATA\_PENALIZARE

Entitatea este dependentă de entitatea PENALIZARE întrucât cheia sa primară este compusă din perechea *cod\_plata\_penalizare* și *cod\_penalizare* (cheia primară a entității PENALIZARE) și reprezintă plata efectuată de un membru a unei penalizări.

* 1. Entitatea ABONAMENT

Entitatea este independentă și reprezintă orice abonament care poate fi cumpărat de un membru. Cheia primară a entității este *cod\_abonament*.

* 1. Entitatea PLATA\_ABONAMENT

Entitatea este independentă și reprezintă orice plată asociată cumpărării unui abonament de către un membru. Cheia primară a entității este *cod\_plata\_abonament*.

1. Descrierea relațiilor
   1. Relația APARTINE

Relația leagă entitățile GEN și CARTE și reprezintă o modalitate de a filtra cărțile din inventar. Cardinalitatea relației este CARTE M(0) – M(1) GEN întrucât o carte trebuie să aparțină minim unui gen, dar poate aparține mai multor genuri, iar unui gen îi pot aparține oricâte cărți.

* 1. Relația SCRIE

Relația leagă entitățile CARTE și AUTOR și reprezintă o modalitate de a corela o carte cu scriitorul sau scriitorii ei. Cardinalitatea relației este AUTOR M(1) – M(0) CARTE întrucât un autor poate scrie mai multe cărți, dar poate fi introdus in baza de date fără ca biblioteca să aibă momentan o carte scrisă de el, iar o carte trebuie scrisă de minim un autor, dar poate fi scrisă de mai mulți.

* 1. Relația IMPRUMUTA

Relația leagă entitățile MEMBRU și CARTE și reprezintă o modalitate de a oferi unui membru posibilitatea de a împrumuta o carte și de a stoca datele despre acest împrumut. Cardinalitatea relației este MEMBRU M(0) – M(0) CARTE întrucât un membru poate împrumuta mai multe cărți, dar nu este necesar să împrumute o carte, iar o carte poate fi împrumutată de oricâți membri, dar nu este necesar să fie împrumutată.

* 1. Relația REZERVA

Relația leagă entitățile MEMBRU și CARTE și reprezintă o modalitate de a oferi unui membru posibilitatea de a rezerva o carte dacă aceasta este împrumutată și de a stoca datele despre această rezervare. Cardinalitatea relației este MEMBRU M(0) – M(0) CARTE întrucât un membru poate rezerva mai multe cărți, dar nu este necesar să rezerve o carte, iar o carte poate fi rezervată de oricâți membri, dar nu este necesar să fie rezervată.

* 1. Relația ARE\_PENALIZARE

Relația leagă entitățile MEMBRU și PENALIZARE și reprezintă o modalitate de a stoca informații despre penalizările primite de un membru al bibliotecii. Cardinalitatea relației este MEMBRU 1 – M(0) PENALIZARE întrucât un membru poate avea mai multe penalizări, dar nu este necesar să aibă o penalizare, iar o penalizare trebuie atribuită unui singur membru.

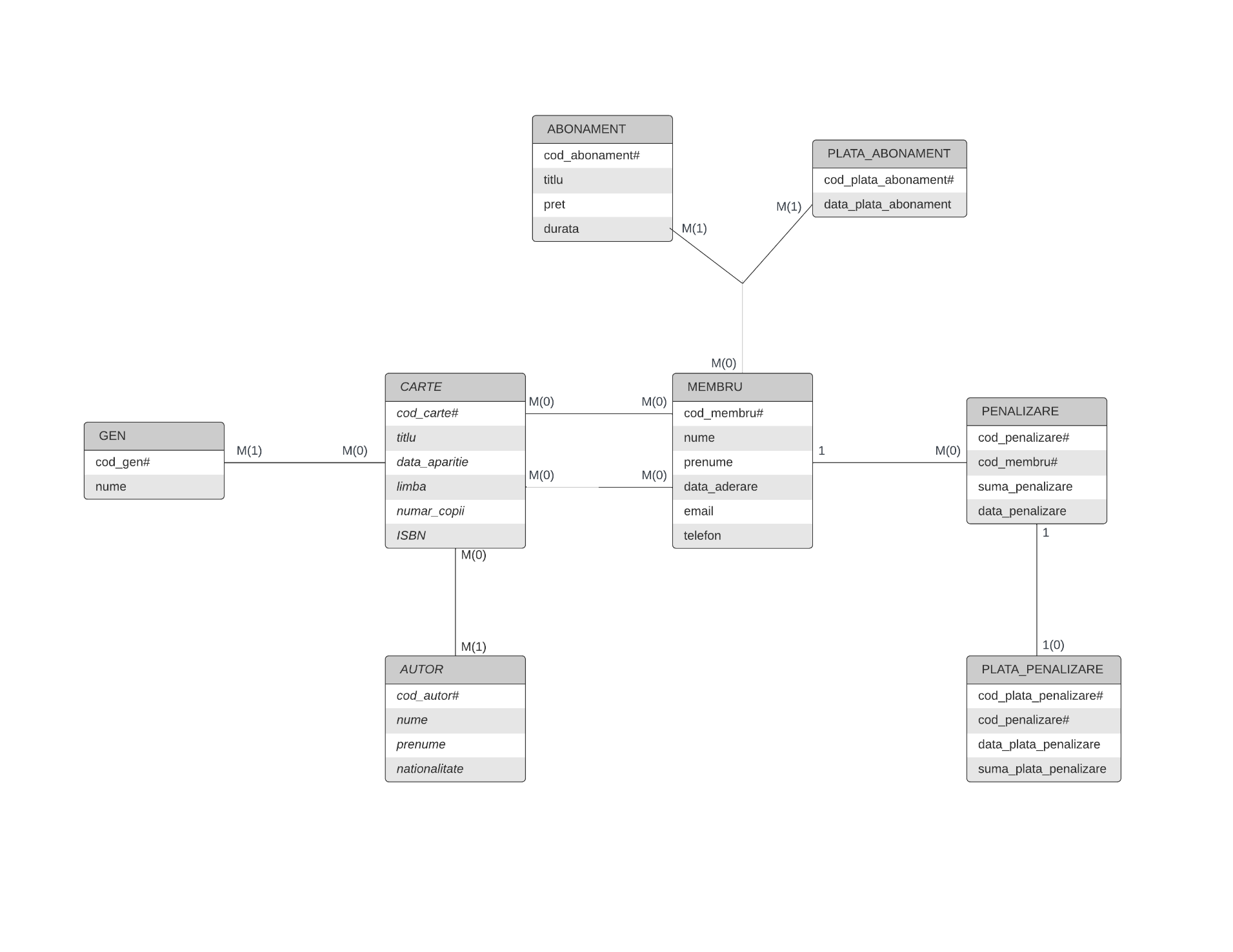
* 1. Relația ARE\_PLATA\_PENALIZARE

Relația leagă entitățile PENALIZARE și PLATA\_PENALIZARE și reprezintă o modalitate de a stoca informațiile despre o penalizare și plata sa, dacă a fost efectuată de membru. Cardinalitatea relației este PENALIZARE 1 – 1(0) PLATA\_PENALIZARE întrucât o penalizare poate fi plătită o singură dată, dar nu trebuie neapărat plătită, iar o plată de penalizare trebuie asociată unei singure penalizări.

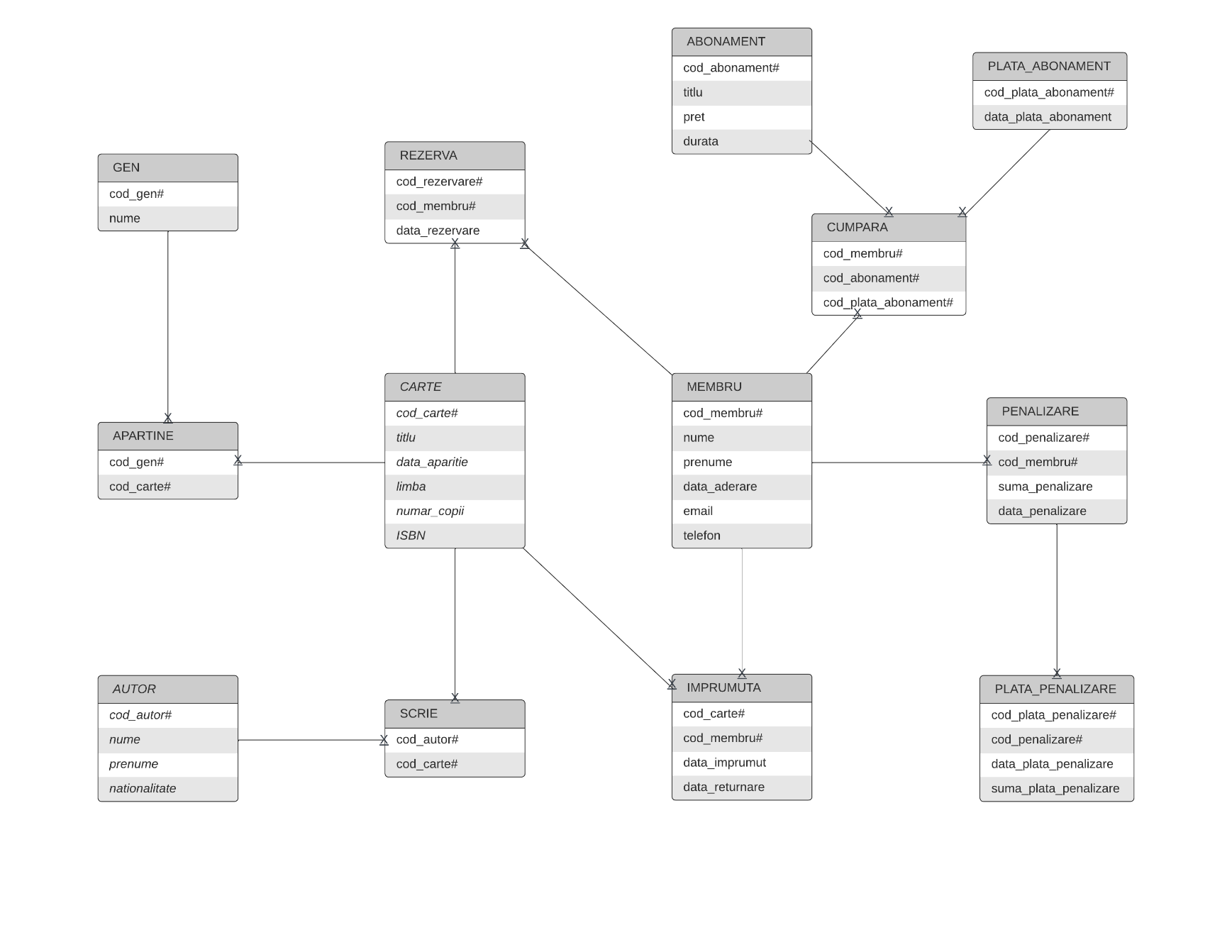
* 1. Relația CUMPARA

Relația leagă entitățile MEMBRU, ABONAMENT și PLATA\_ABONAMENT și reprezintă o modalitate de a stoca informațiile despre abonamentele cumpărate de un membru si plățile asociate acestor plăți. Cardinalitatea relației este MEMBRU M(0), ABONAMENT M(1), PLATA\_ABONAMENT M(1) întrucât un membru poate cumpăra mai multe abonamente cărora le vor fi asociate mai multe plăți, dar trebuie să cumpere cel puțin un abonament căruia îi va fi asociat o singură plată, un abonament poate fi cumpărat de mai mulți membri cărora le vor fi asociate mai multe plăți per membru, dar nu este necesar să fie cumpărat de un membru sau asociat unei plăți, iar o plata de abonament poate fi asociată mai multor membri si abonamente, dar nu e este necesar.

1. Descrierea atributelor
   1. Tabela CARTE
2. Atributul *cod\_carte* este de tipul NUMBER și trebuie să fie unic și diferit de null întrucât este cheie primară.
3. Atributul *titlu* este de tipul VARCHAR2 și nu trebuie să fie null.
4. Atributul *data\_aparitie* este de tipul DATE și nu trebuie să fie null
5. Atributul *limba* este de tipul VARCHAR2 și are valoarea implicită “romana”.
6. Atributul *numar\_copii* este de tipul NUMBER și are valoarea implicită 1.
7. Atributul *ISBN* este de tipul VARCHAR2 și nu trebuie să fie null.
   1. Tabela AUTOR
8. Atributul *cod\_autor* este de tipul NUMBER și trebuie să fie unic și diferit de null întrucât este cheie primară.
9. Atributul *nume* este de tipul VARCHAR2 și trebuie să fie diferit de null.
10. Atributul *prenume* este de tipul VARCHAR2.
11. Atributul *nationalitate* este de tipul VARCHAR2 și are valoarea implicită “roman”.
    1. Tabela GEN
12. Atributul *cod\_gen* este de tipul NUMBER și trebuie să fie unic și diferit de null întrucât este cheie primară.
13. Atributul *nume* este de tipul VARCHAR2 și trebuie să fie diferit de null.
    1. Tabela APARTINE
14. Atributul *cod\_gen* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
15. Atributul *cod\_carte* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
16. Cheia primară este alcătuită din atributele *cod\_carte* și *cod\_gen*.
    1. Tabela SCRIE
17. Atributul *cod\_autor* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
18. Atributul *cod\_carte* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
19. Cheia primară este alcătuită din atributele *cod\_carte* si *cod\_autor*.
    1. Tabela MEMBRU
20. Atributul *cod\_membru* este de tipul NUMBER și trebuie să fie unic și diferit de null întrucât este cheie primară.
21. Atributul *nume* este de tipul VARCHAR2 și trebuie să fie diferit de null.
22. Atributul *prenume* este de tipul VARCHAR2.
23. Atributul *data\_aderare* este de tipul DATE și trebuie să fie diferit de null.
24. Atributul *email* este de tipul VARCHAR2.
25. Atributul *telefon* este de tipul VARCHAR2.
    1. Tabela REZERVA
26. Atributul *cod\_carte* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
27. Atributul *cod\_membru* este de tipul NUMBER și cheie externă cu **on delete cascade** și diferit de null.
28. Atributul *data\_rezervare* este de tipul DATE și este diferit de null.
29. Cheia primară este alcătuită din atributele *cod\_carte* și *cod\_membru*.
    1. Tabela IMPRUMUTA
30. Atributul *cod\_carte* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
31. Atributul *cod\_membru* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
32. Atributul *data\_imprumut* este de tipul DATE și are valoare implicită data curentă.
33. Atributul *data\_returnare* este de tipul DATE și trebuie să fie mai mare sau egal cu *data\_imprumut*.
34. Cheia primară este alcătuită din atributele *cod\_carte* și *cod\_membru*.
    1. Tabela PENALIZARE
35. Atributul *cod\_penalizare* este de tipul NUMBER și este diferit de null întrucât este cheie primară.
36. Atributul *cod\_membru* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
37. Atributul *suma\_penalizare* este de tipul NUMBER și trebuie să fie mai mare decât 0 și diferit de null.
38. Atributul *data\_penalizare* este de tipul DATE și are valoare implicită data curentă.
    1. Tabela PLATA\_PENALIZARE
39. Atributul *cod\_plata\_penalizare* este de tipul NUMBER și este diferit de null întrucât este cheie primară.
40. Atributul *cod\_penalizare* este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
41. Atributul *data\_plata\_penalizare* este de tipul DATE și are valoare implicită data curentă.
42. Atributul *suma\_plata\_penalizare* este de tipul NUMBER și trebuie să fie mai mare decât 0 și diferit de null.
    1. Tabela ABONAMENT
43. Atributul *cod\_abonament* este de tipul NUMBER și trebuie să fie diferit de null întrucât este cheie primară.
44. Atributul *titlu* este de tipul VARCHAR2 și este diferit de null.
45. Atributul *pret* este de tipul NUMBER și trebuie să fie mai mare decât 0 și diferit de null.
46. Atributul *durata* este de tipul NUMBER și trebuie să fie diferit de null și mai mare decât 0.
    1. Tabela PLATA\_ABONAMENT
47. Atributul *cod\_plata\_abonament* este de tipul NUMBER și trebuie să fie diferit de null întrucât este cheie primară.
48. Atributul *data\_plata\_abonament* este de tipul DATE și are valoarea implicită data curentă.
    1. Tabela CUMPARA
49. Atributul cod\_membru este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
50. Atributul cod\_abonament este de tipul NUMBER și este cheie externă cu **on delete cascade** și diferit de null.
51. Atributul cod\_plata\_abonament este de tipul NUMBER și este cheie externă cu **on delete cascade**.
52. Cheia primară este alcătuită din atributele *cod\_membru*, *cod\_abonament* și *cod\_plata\_abonament*.
53. Diagrama Entitate-Relație



1. Diagrama Conceptuală



1. Schemele relaționale corespunzătoare diagramei conceptuale

MEMBRU(cod\_membru#, nume, prenume, data\_aderare, email, telefon)

CARTE(cod\_carte#, titlu, data\_aparitie, limba, numar\_copii, ISBN)

AUTOR(cod\_autor#, nume, prenume, nationalitate)

PENALIZARE(cod\_penalizare#, cod\_membru#, data\_penalizare, suma\_penalizare)

PLATA\_PENALIZARE(cod\_plata\_penalizare#, cod\_penalizare#, data\_plata\_penalizare, suma\_plata\_penalizare)

ABONAMENT(cod\_abonament#, titlu, pret, durata)

PLATA\_ABONAMENT(cod\_plata\_abonament#, data\_plata\_abonament)

GEN(cod\_gen#, nume)

CUMPARA(cod\_membru#, cod\_abonament#, cod\_plata\_abonament#)

SCRIE(cod\_carte#, cod\_autor#)

APARTINE(cod\_carte#, cod\_gen#)

IMPRUMUTA(cod\_carte#, cod\_membru#, data\_imprumut, data\_returnare)

REZERVA(cod\_carte#, cod\_membru#, data\_rezervare)

1. Formele normale
   1. FN1

Preuspunem că tabelele CARTE și AUTOR nu ar fi separate și am avea tabela CARTE\_AUTOR cu următoarea schemă relațională:

CARTE\_AUTOR(cod\_carte#, titlu, data\_aparitie, limba, numar\_copii, ISBN, cod\_autor#, nume, prenume, nationalitate)

Redundanța datelor este mare, pentru fiecare carte scrisă de acest autor pe care o introducem, va trebui să reintroducem și datele despre autor. O problemă poate apărea și dacă ștergem toate cărțile scrise de acest autor întrucât vom pierde informațiile despre el. Astfel soluția este împărțirea datelor în două tabele, AUTOR și CARTE ce vor avea relația SCRIE între ele ca în diagramă.

* 1. FN2

Presupunem că în tabela PENALIZARE am avea și atributul nume\_membru, schema relațională fiind:

PENALIZARE(cod\_membru#, cod\_penalizare#, suma\_penalizare, data\_penalizare, nume\_membru)

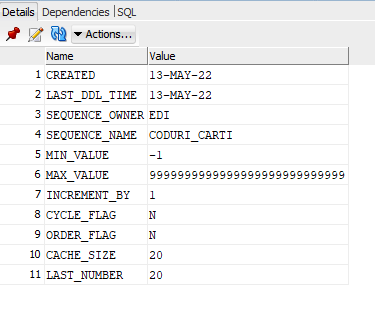
Acest atribut depinde doar de cod\_membru#, ce reprezintă doar o parte din cheia primară a tabelei PENALIZARE. Soluția este să ștergem atributul cu totul întrucât acesta poate fi obținut cu ajutorului tabelei MEMBRU.

* 1. FN3

Presupunem că în tabela MEMBRU am avea și atributul nr\_carti\_imprumutate, schema relațională fiind:

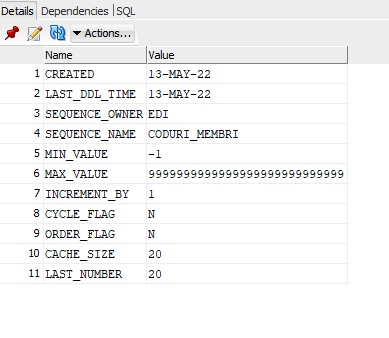
MEMBRU(cod\_membru#, nume, prenume, data\_aderare, email, telefon, nr\_carti\_imprumutate)

Acest atribut ar trebui să depindă doar de cheia primară cod\_membru# ca diagrama să rămână în FN3, dar el depinde și de tabela IMPRUMUTA întrucât valoarea acestui atribut ar fi egală cu numărul de apariții al atributului cod\_membru în această tabelă. Astfel, soluția este să eliminăm atributul pentru a elimina odată cu el și dependența după cum se poate observa în diagrama conceptuală curentă.



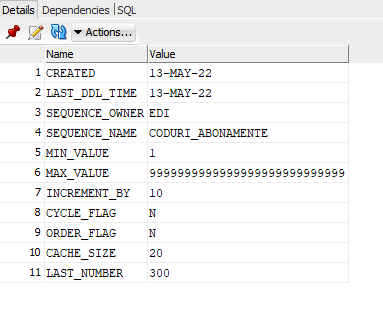
1. Secvențele pentru inserare
   1. Secvența pentru tabela CARTE

CREATE SEQUENCE "EDI"."CODURI\_CARTI" MINVALUE -1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 20 CACHE 20 NOORDER NOCYCLE ;



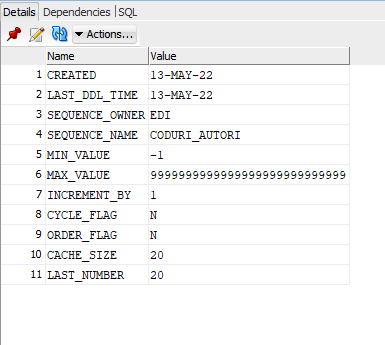
* 1. Secvența pentru tabela MEMBRU

CREATE SEQUENCE "EDI"."CODURI\_MEMBRI" MINVALUE -1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 20 CACHE 20 NOORDER NOCYCLE ;

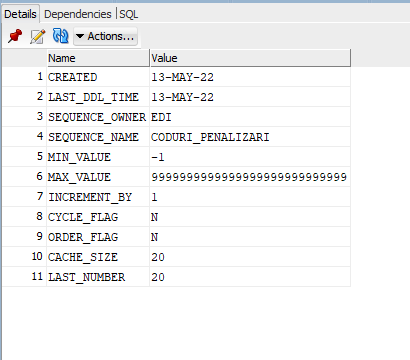


* 1. Secvența pentru tabela ABONAMENT

CREATE SEQUENCE "EDI"."CODURI\_ABONAMENTE" MINVALUE 1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 10 START WITH 300 CACHE 20 NOORDER NOCYCLE ;

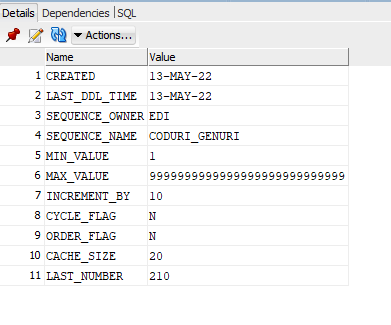
* 1. Secvența pentru tabela AUTOR

CREATE SEQUENCE "EDI"."CODURI\_AUTORI" MINVALUE -1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 20 CACHE 20 NOORDER NOCYCLE ;



* 1. Secvența pentru tabela PENALIZARE

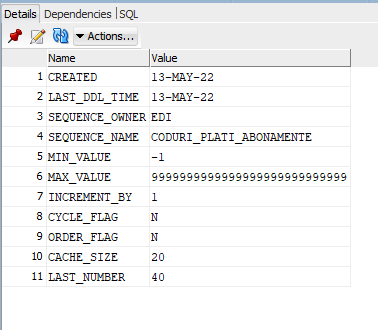
CREATE SEQUENCE "EDI"."CODURI\_PENALIZARI" MINVALUE -1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 20 CACHE 20 NOORDER NOCYCLE ;

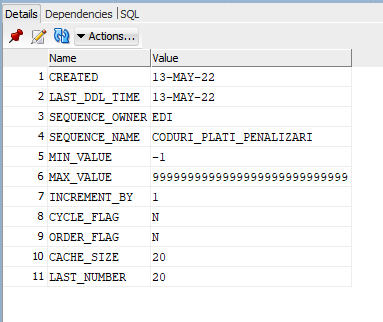


* 1. Secvența pentru tabela GEN

CREATE SEQUENCE "EDI"."CODURI\_GENURI" MINVALUE 1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 10 START WITH 210 CACHE 20 NOORDER NOCYCLE ;

* 1. Secvența pentru tabela PLATA\_ABONAMENT

CREATE SEQUENCE "EDI"."CODURI\_PLATI\_ABONAMENTE" MINVALUE -1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 40 CACHE 20 NOORDER NOCYCLE ;

* 1. Secvența pentru tabela PLATA\_PENALIZARE

CREATE SEQUENCE "EDI"."CODURI\_PLATI\_PENALIZARI" MINVALUE -1 MAXVALUE 9999999999999999999999999999 INCREMENT BY 1 START WITH 20 CACHE 20 NOORDER NOCYCLE ;

1. Crearea tabelelor și inserarea datelor
   1. Tabela CARTE

CREATE TABLE "EDI"."CARTE"

( "COD\_CARTE" NUMBER(6,0),

"TITLU" VARCHAR2(40 BYTE),

"DATA\_APARITIE" DATE,

"LIMBA" VARCHAR2(20 BYTE) DEFAULT 'romana',

"NUMAR\_COPII" NUMBER(3,0) DEFAULT 1,

"ISBN" VARCHAR2(13 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.CARTE

SET DEFINE OFF;

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (13,'Moartea lui Ivan Ilici',to\_date('21-OCT-86','DD-MON-RR'),'rusa',1,'9789734634316');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (1,'Ion',to\_date('20-NOV-20','DD-MON-RR'),'romana',1,'9780805756951');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (2,'Moara cu Noroc',to\_date('21-OCT-81','DD-MON-RR'),'romana',1,'9780483094147');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (3,'Enigma Otiliei',to\_date('21-OCT-38','DD-MON-RR'),'romana',1,'9789731973173');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (4,'Morometii vol. 1',to\_date('21-OCT-55','DD-MON-RR'),'romana',3,'5948486009238');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (5,'Morometii vol. 2',to\_date('21-OCT-67','DD-MON-RR'),'romana',3,'9789736698422');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (7,'Ecce Homo',to\_date('21-OCT-08','DD-MON-RR'),'germana',2,'9780140445152');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (8,'Adam si Eva',to\_date('21-OCT-25','DD-MON-RR'),'romana',2,'9789736756139');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (9,'Anna Karenina',to\_date('21-OCT-78','DD-MON-RR'),'rusa',4,'9781853262715');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (11,'365 de zile care au marcat istoria',to\_date('10-JAN-21','DD-MON-RR'),'romana',1,'9786066001052');

Insert into EDI.CARTE (COD\_CARTE,TITLU,DATA\_APARITIE,LIMBA,NUMAR\_COPII,ISBN) values (12,'Animal Farm',to\_date('17-AUG-45','DD-MON-RR'),'engleza',2,'0141036133');

--------------------------------------------------------

-- DDL for Index PK\_CARTE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_CARTE" ON "EDI"."CARTE" ("COD\_CARTE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table CARTE

--------------------------------------------------------

ALTER TABLE "EDI"."CARTE" ADD CONSTRAINT "PK\_CARTE" PRIMARY KEY ("COD\_CARTE")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

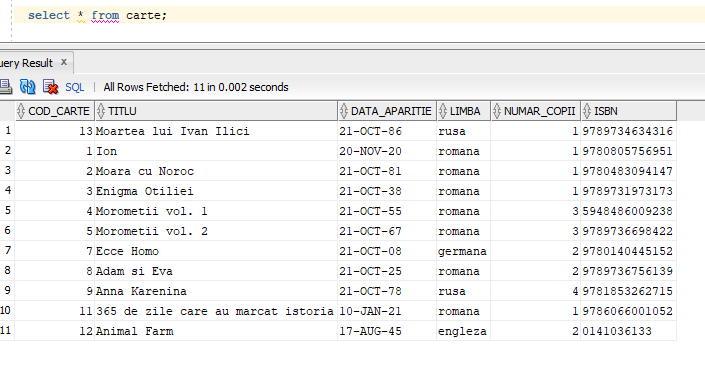
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

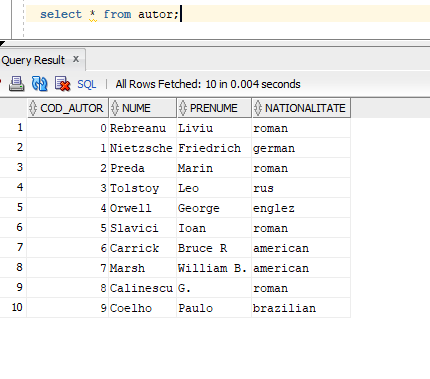
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."CARTE" MODIFY ("ISBN" CONSTRAINT "NULL\_ISBN" NOT NULL ENABLE);

ALTER TABLE "EDI"."CARTE" MODIFY ("DATA\_APARITIE" CONSTRAINT "NULL\_DATA\_APARITIE" NOT NULL ENABLE);

 ALTER TABLE "EDI"."CARTE" MODIFY ("TITLU" CONSTRAINT "NULL\_TITLU" NOT NULL ENABLE);

* 1.  Tabela AUTOR

CREATE TABLE "EDI"."AUTOR"

( "COD\_AUTOR" NUMBER(6,0),

"NUME" VARCHAR2(20 BYTE),

"PRENUME" VARCHAR2(20 BYTE),

"NATIONALITATE" VARCHAR2(20 BYTE) DEFAULT 'roman'

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.AUTOR

SET DEFINE OFF;

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (0,'Rebreanu','Liviu','roman');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (1,'Nietzsche','Friedrich','german');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (2,'Preda','Marin','roman');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (3,'Tolstoy','Leo','rus');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (4,'Orwell','George','englez');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (5,'Slavici','Ioan','roman');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (6,'Carrick','Bruce R','american');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (7,'Marsh','William B.','american');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (8,'Calinescu','G.','roman');

Insert into EDI.AUTOR (COD\_AUTOR,NUME,PRENUME,NATIONALITATE) values (9,'Coelho','Paulo','brazilian');

--------------------------------------------------------

-- DDL for Index PK\_AUTOR

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_AUTOR" ON "EDI"."AUTOR" ("COD\_AUTOR")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table AUTOR

--------------------------------------------------------

ALTER TABLE "EDI"."AUTOR" ADD CONSTRAINT "PK\_AUTOR" PRIMARY KEY ("COD\_AUTOR")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."AUTOR" MODIFY ("NUME" CONSTRAINT "NULL\_NUME\_AUTOR" NOT NULL ENABLE);

* 1. Tabela MEMBRU

CREATE TABLE "EDI"."MEMBRU"

( "COD\_MEMBRU" NUMBER(6,0),

"NUME" VARCHAR2(20 BYTE),

"PRENUME" VARCHAR2(20 BYTE),

"DATA\_ADERARE" DATE,

"EMAIL" VARCHAR2(40 BYTE),

"TELEFON" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.MEMBRU

SET DEFINE OFF;

Insert into EDI.MEMBRU (COD\_MEMBRU,NUME,PRENUME,DATA\_ADERARE,EMAIL,TELEFON) values (0,'Sabau','Eduard',to\_date('05-MAR-18','DD-MON-RR'),'sabaueduard21@yahoo.com','0712345678');

Insert into EDI.MEMBRU (COD\_MEMBRU,NUME,PRENUME,DATA\_ADERARE,EMAIL,TELEFON) values (1,'Sabau','David',to\_date('06-MAR-18','DD-MON-RR'),'sabaudavid@yahoo.com','0712321678');

Insert into EDI.MEMBRU (COD\_MEMBRU,NUME,PRENUME,DATA\_ADERARE,EMAIL,TELEFON) values (2,'Rosu','Ilinca',to\_date('30-MAY-20','DD-MON-RR'),'rosuilinca@yahoo.com','0712545678');

Insert into EDI.MEMBRU (COD\_MEMBRU,NUME,PRENUME,DATA\_ADERARE,EMAIL,TELEFON) values (3,'Paun','Tudor',to\_date('19-NOV-19','DD-MON-RR'),'pauntudor@gmail.com','0712343478');

Insert into EDI.MEMBRU (COD\_MEMBRU,NUME,PRENUME,DATA\_ADERARE,EMAIL,TELEFON) values (4,'Stanescu','Lata',to\_date('01-JAN-21','DD-MON-RR'),'stanesculata@yahoo.com','0712345678');

--------------------------------------------------------

-- DDL for Index PK\_MEMBRU

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_MEMBRU" ON "EDI"."MEMBRU" ("COD\_MEMBRU")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table MEMBRU

--------------------------------------------------------

ALTER TABLE "EDI"."MEMBRU" ADD CONSTRAINT "PK\_MEMBRU" PRIMARY KEY ("COD\_MEMBRU")

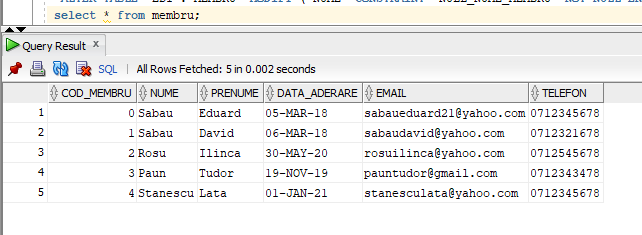
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

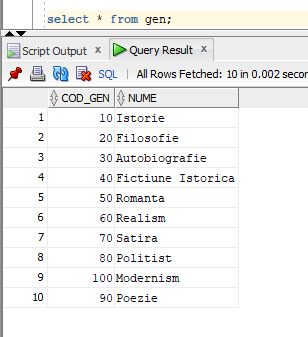
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."MEMBRU" MODIFY ("DATA\_ADERARE" CONSTRAINT "NULL\_DATA\_ADERARE" NOT NULL ENABLE);

 ALTER TABLE "EDI"."MEMBRU" MODIFY ("NUME" CONSTRAINT "NULL\_NUME\_MEMBRU" NOT NULL ENABLE);

* 1.  Tabela GEN

CREATE TABLE "EDI"."GEN"

( "COD\_GEN" NUMBER(6,0),

"NUME" VARCHAR2(20 BYTE)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.GEN

SET DEFINE OFF;

Insert into EDI.GEN (COD\_GEN,NUME) values (10,'Istorie');

Insert into EDI.GEN (COD\_GEN,NUME) values (20,'Filosofie');

Insert into EDI.GEN (COD\_GEN,NUME) values (30,'Autobiografie');

Insert into EDI.GEN (COD\_GEN,NUME) values (40,'Fictiune Istorica');

Insert into EDI.GEN (COD\_GEN,NUME) values (50,'Romanta');

Insert into EDI.GEN (COD\_GEN,NUME) values (60,'Realism');

Insert into EDI.GEN (COD\_GEN,NUME) values (70,'Satira');

Insert into EDI.GEN (COD\_GEN,NUME) values (80,'Politist');

Insert into EDI.GEN (COD\_GEN,NUME) values (100,'Modernism');

Insert into EDI.GEN (COD\_GEN,NUME) values (90,'Poezie');

--------------------------------------------------------

-- DDL for Index PK\_GEN

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_GEN" ON "EDI"."GEN" ("COD\_GEN")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table GEN

--------------------------------------------------------

ALTER TABLE "EDI"."GEN" ADD CONSTRAINT "PK\_GEN" PRIMARY KEY ("COD\_GEN")

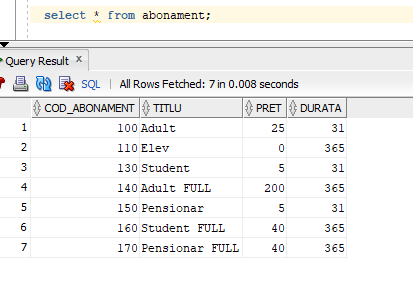
USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."GEN" MODIFY ("NUME" CONSTRAINT "NULL\_NUME\_GEN" NOT NULL ENABLE);

* 1.  Tabela ABONAMENT

CREATE TABLE "EDI"."ABONAMENT"

( "COD\_ABONAMENT" NUMBER(6,0),

"TITLU" VARCHAR2(20 BYTE),

"PRET" NUMBER(8,2),

"DURATA" NUMBER(7,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.ABONAMENT

SET DEFINE OFF;

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (100,'Adult',25,31);

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (110,'Elev',0,365);

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (130,'Student',5,31);

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (140,'Adult FULL',200,365);

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (150,'Pensionar',5,31);

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (160,'Student FULL',40,365);

Insert into EDI.ABONAMENT (COD\_ABONAMENT,TITLU,PRET,DURATA) values (170,'Pensionar FULL',40,365);

--------------------------------------------------------

-- DDL for Index PK\_ABONAMENT

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_ABONAMENT" ON "EDI"."ABONAMENT" ("COD\_ABONAMENT")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table ABONAMENT

--------------------------------------------------------

ALTER TABLE "EDI"."ABONAMENT" ADD CONSTRAINT "PK\_ABONAMENT" PRIMARY KEY ("COD\_ABONAMENT")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

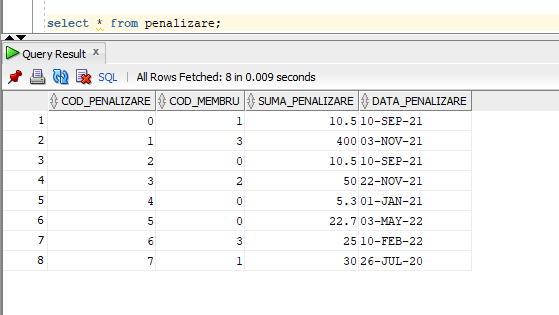
TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."ABONAMENT" ADD CONSTRAINT "CK\_ABONAMENT\_DURATA" CHECK (durata > 0) ENABLE;

ALTER TABLE "EDI"."ABONAMENT" MODIFY ("DURATA" CONSTRAINT "NULL\_ABONAMENT\_DURATA" NOT NULL ENABLE);

ALTER TABLE "EDI"."ABONAMENT" MODIFY ("PRET" CONSTRAINT "NULL\_ABONAMENT\_PRET" NOT NULL ENABLE);

ALTER TABLE "EDI"."ABONAMENT" MODIFY ("TITLU" CONSTRAINT "NULL\_ABONAMENT\_TITLU" NOT NULL ENABLE);

* 1.  Tabela PENALIZARE

CREATE TABLE "EDI"."PENALIZARE"

( "COD\_PENALIZARE" NUMBER(6,0),

"COD\_MEMBRU" NUMBER(6,0),

"SUMA\_PENALIZARE" NUMBER(8,2),

"DATA\_PENALIZARE" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.PENALIZARE

SET DEFINE OFF;

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (0,1,10.5,to\_date('10-SEP-21','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (1,3,400,to\_date('03-NOV-21','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (2,0,10.5,to\_date('10-SEP-21','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (3,2,50,to\_date('22-NOV-21','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (4,0,5.3,to\_date('01-JAN-21','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (5,0,22.7,to\_date('03-MAY-22','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (6,3,25,to\_date('10-FEB-22','DD-MON-RR'));

Insert into EDI.PENALIZARE (COD\_PENALIZARE,COD\_MEMBRU,SUMA\_PENALIZARE,DATA\_PENALIZARE) values (7,1,30,to\_date('26-JUL-20','DD-MON-RR'));

--------------------------------------------------------

-- DDL for Index PK\_PENALIZARE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_PENALIZARE" ON "EDI"."PENALIZARE" ("COD\_PENALIZARE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table PENALIZARE

--------------------------------------------------------

ALTER TABLE "EDI"."PENALIZARE" ADD CONSTRAINT "PK\_PENALIZARE" PRIMARY KEY ("COD\_PENALIZARE")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."PENALIZARE" ADD CONSTRAINT "CK\_SUMA\_PENALIZARE" CHECK (suma\_penalizare > 0) ENABLE;

ALTER TABLE "EDI"."PENALIZARE" MODIFY ("SUMA\_PENALIZARE" CONSTRAINT "NULL\_SUMA\_PENALIZARE" NOT NULL ENABLE);

--------------------------------------------------------

-- Ref Constraints for Table PENALIZARE

--------------------------------------------------------

ALTER TABLE "EDI"."PENALIZARE" ADD CONSTRAINT "FK\_PENALIZARE\_MEMBRU" FOREIGN KEY ("COD\_MEMBRU")

REFERENCES "EDI"."MEMBRU" ("COD\_MEMBRU") ON DELETE CASCADE ENABLE;

* 1. Tabela PLATA\_PENALIZARE

CREATE TABLE "EDI"."PLATA\_PENALIZARE"

( "COD\_PLATA\_PENALIZARE" NUMBER(6,0),

"COD\_PENALIZARE" NUMBER(6,0),

"DATA\_PLATA\_PENALZARE" DATE DEFAULT sysdate,

"SUMA\_PLATA\_PENALIZARE" NUMBER(8,2)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.PLATA\_PENALIZARE

SET DEFINE OFF;

Insert into EDI.PLATA\_PENALIZARE (COD\_PLATA\_PENALIZARE,COD\_PENALIZARE,DATA\_PLATA\_PENALZARE,SUMA\_PLATA\_PENALIZARE) values (0,0,to\_date('21-SEP-21','DD-MON-RR'),10.5);

Insert into EDI.PLATA\_PENALIZARE (COD\_PLATA\_PENALIZARE,COD\_PENALIZARE,DATA\_PLATA\_PENALZARE,SUMA\_PLATA\_PENALIZARE) values (1,5,to\_date('04-MAY-22','DD-MON-RR'),22.7);

Insert into EDI.PLATA\_PENALIZARE (COD\_PLATA\_PENALIZARE,COD\_PENALIZARE,DATA\_PLATA\_PENALZARE,SUMA\_PLATA\_PENALIZARE) values (2,3,to\_date('21-SEP-22','DD-MON-RR'),50);

Insert into EDI.PLATA\_PENALIZARE (COD\_PLATA\_PENALIZARE,COD\_PENALIZARE,DATA\_PLATA\_PENALZARE,SUMA\_PLATA\_PENALIZARE) values (3,2,to\_date('10-OCT-21','DD-MON-RR'),10.5);

Insert into EDI.PLATA\_PENALIZARE (COD\_PLATA\_PENALIZARE,COD\_PENALIZARE,DATA\_PLATA\_PENALZARE,SUMA\_PLATA\_PENALIZARE) values (4,1,to\_date('04-NOV-21','DD-MON-RR'),400);

--------------------------------------------------------

-- DDL for Index PK\_PLATA\_PENALIZARE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_PLATA\_PENALIZARE" ON "EDI"."PLATA\_PENALIZARE" ("COD\_PLATA\_PENALIZARE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table PLATA\_PENALIZARE

--------------------------------------------------------

ALTER TABLE "EDI"."PLATA\_PENALIZARE" ADD CONSTRAINT "PK\_PLATA\_PENALIZARE" PRIMARY KEY ("COD\_PLATA\_PENALIZARE")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."PLATA\_PENALIZARE" ADD CONSTRAINT "CK\_SUMA\_PLATA\_PENALIZARE" CHECK (suma\_plata\_penalizare > 0) ENABLE;

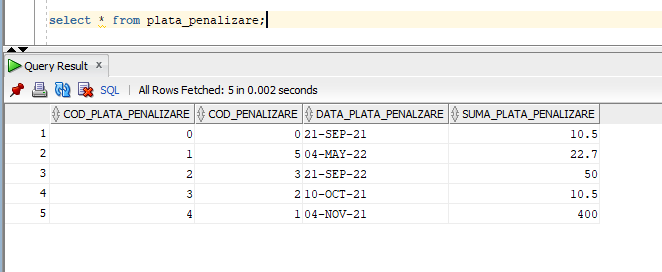
ALTER TABLE "EDI"."PLATA\_PENALIZARE" MODIFY ("SUMA\_PLATA\_PENALIZARE" CONSTRAINT "NULL\_SUMA\_PLATA\_PENALIZARE" NOT NULL ENABLE);

--------------------------------------------------------

-- Ref Constraints for Table PLATA\_PENALIZARE

--------------------------------------------------------

ALTER TABLE "EDI"."PLATA\_PENALIZARE" ADD CONSTRAINT "FK\_PLATA\_PENALIZARE\_PENALIZARE" FOREIGN KEY ("COD\_PENALIZARE")

 REFERENCES "EDI"."PENALIZARE" ("COD\_PENALIZARE") ON DELETE CASCADE ENABLE;

* 1. Tabela PLATA\_ABONAMENT

CREATE TABLE "EDI"."PLATA\_ABONAMENT"

( "COD\_PLATA\_ABONAMENT" NUMBER(6,0),

"DATA\_PLATA\_ABONAMENT" DATE DEFAULT sysdate

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.PLATA\_ABONAMENT

SET DEFINE OFF;

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (10,to\_date('01-JAN-21','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (11,to\_date('19-NOV-22','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (12,to\_date('19-NOV-21','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (13,to\_date('19-NOV-20','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (14,to\_date('19-NOV-19','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (15,to\_date('30-MAY-22','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (16,to\_date('30-MAY-21','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (17,to\_date('30-MAY-20','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (18,to\_date('06-FEB-22','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (19,to\_date('06-FEB-21','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (20,to\_date('06-FEB-20','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (21,to\_date('06-FEB-19','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (22,to\_date('06-FEB-18','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (23,to\_date('05-FEB-22','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (24,to\_date('05-FEB-21','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (25,to\_date('05-FEB-20','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (26,to\_date('05-FEB-19','DD-MON-RR'));

Insert into EDI.PLATA\_ABONAMENT (COD\_PLATA\_ABONAMENT,DATA\_PLATA\_ABONAMENT) values (27,to\_date('05-FEB-18','DD-MON-RR'));

--------------------------------------------------------

-- DDL for Index PK\_PLATA\_ABONAMNET

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_PLATA\_ABONAMNET" ON "EDI"."PLATA\_ABONAMENT" ("COD\_PLATA\_ABONAMENT")

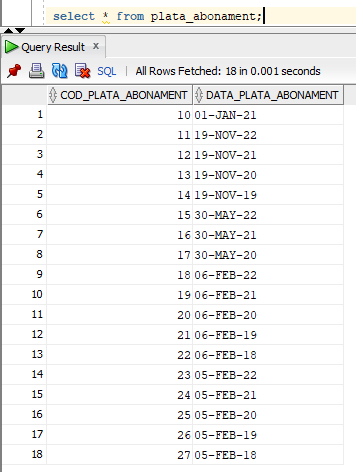
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table PLATA\_ABONAMENT

--------------------------------------------------------

ALTER TABLE "EDI"."PLATA\_ABONAMENT" ADD CONSTRAINT "PK\_PLATA\_ABONAMNET" PRIMARY KEY ("COD\_PLATA\_ABONAMENT")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

* 1. Tabela IMPRUMUTA

CREATE TABLE "EDI"."IMPRUMUTA"

( "COD\_CARTE" NUMBER(6,0),

"COD\_MEMBRU" NUMBER(6,0),

"DATA\_IMPRUMUT" DATE DEFAULT sysdate,

"DATA\_RETURNARE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.IMPRUMUTA

SET DEFINE OFF;

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (13,2,to\_date('19-APR-21','DD-MON-RR'),to\_date('28-APR-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (1,0,to\_date('21-OCT-21','DD-MON-RR'),to\_date('30-OCT-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (2,0,to\_date('06-MAY-19','DD-MON-RR'),to\_date('06-JUN-19','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (7,0,to\_date('18-DEC-20','DD-MON-RR'),to\_date('01-JAN-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (2,1,to\_date('10-FEB-21','DD-MON-RR'),to\_date('19-FEB-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (2,3,to\_date('29-AUG-21','DD-MON-RR'),to\_date('10-SEP-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (12,3,to\_date('21-NOV-20','DD-MON-RR'),to\_date('10-DEC-20','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (9,2,to\_date('11-NOV-21','DD-MON-RR'),to\_date('22-NOV-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (11,0,to\_date('21-OCT-21','DD-MON-RR'),to\_date('30-OCT-21','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (8,0,to\_date('21-OCT-19','DD-MON-RR'),to\_date('02-NOV-19','DD-MON-RR'));

Insert into EDI.IMPRUMUTA (COD\_CARTE,COD\_MEMBRU,DATA\_IMPRUMUT,DATA\_RETURNARE) values (1,3,to\_date('03-NOV-20','DD-MON-RR'),to\_date('03-NOV-21','DD-MON-RR'));

--------------------------------------------------------

-- DDL for Index PK\_IMPRUMUTA\_MEMBRU\_CARTE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_IMPRUMUTA\_MEMBRU\_CARTE" ON "EDI"."IMPRUMUTA" ("COD\_CARTE", "COD\_MEMBRU")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table IMPRUMUTA

--------------------------------------------------------

ALTER TABLE "EDI"."IMPRUMUTA" ADD CONSTRAINT "PK\_IMPRUMUTA\_MEMBRU\_CARTE" PRIMARY KEY ("COD\_CARTE", "COD\_MEMBRU")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

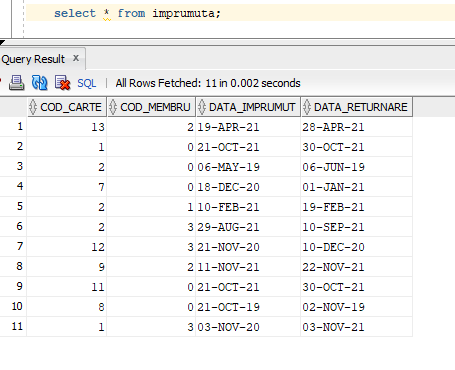
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."IMPRUMUTA" ADD CONSTRAINT "CK\_IMPRUMUTA\_DATA\_RETURNARE" CHECK (data\_returnare >= data\_imprumut) ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table IMPRUMUTA

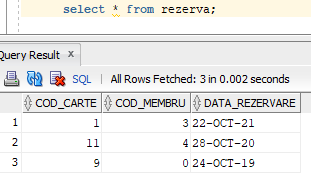
--------------------------------------------------------

ALTER TABLE "EDI"."IMPRUMUTA" ADD CONSTRAINT "FK\_IMPRUMUTA\_CARTE" FOREIGN KEY ("COD\_CARTE")

REFERENCES "EDI"."CARTE" ("COD\_CARTE") ON DELETE CASCADE ENABLE;

ALTER TABLE "EDI"."IMPRUMUTA" ADD CONSTRAINT "FK\_IMPRUMUTA\_MEMBRU" FOREIGN KEY ("COD\_MEMBRU")

REFERENCES "EDI"."MEMBRU" ("COD\_MEMBRU") ON DELETE CASCADE ENABLE;

* 1. Tabela REZERVA

CREATE TABLE "EDI"."REZERVA"

( "COD\_CARTE" NUMBER(6,0),

"COD\_MEMBRU" NUMBER(6,0),

"DATA\_REZERVARE" DATE

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.REZERVA

SET DEFINE OFF;

Insert into EDI.REZERVA (COD\_CARTE,COD\_MEMBRU,DATA\_REZERVARE) values (1,3,to\_date('22-OCT-21','DD-MON-RR'));

Insert into EDI.REZERVA (COD\_CARTE,COD\_MEMBRU,DATA\_REZERVARE) values (11,4,to\_date('28-OCT-20','DD-MON-RR'));

Insert into EDI.REZERVA (COD\_CARTE,COD\_MEMBRU,DATA\_REZERVARE) values (9,0,to\_date('24-OCT-19','DD-MON-RR'));

--------------------------------------------------------

-- DDL for Index PK\_REZERVA\_MEMBRU\_CARTE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_REZERVA\_MEMBRU\_CARTE" ON "EDI"."REZERVA" ("COD\_CARTE", "COD\_MEMBRU")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table REZERVA

--------------------------------------------------------

ALTER TABLE "EDI"."REZERVA" ADD CONSTRAINT "PK\_REZERVA\_MEMBRU\_CARTE" PRIMARY KEY ("COD\_CARTE", "COD\_MEMBRU")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

ALTER TABLE "EDI"."REZERVA" MODIFY ("DATA\_REZERVARE" CONSTRAINT "NULL\_DATA\_REZERVARE" NOT NULL ENABLE);

--------------------------------------------------------

-- Ref Constraints for Table REZERVA

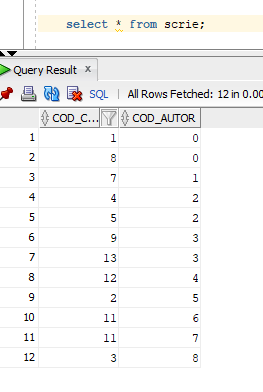
--------------------------------------------------------

ALTER TABLE "EDI"."REZERVA" ADD CONSTRAINT "FK\_REZERVA\_CARTE" FOREIGN KEY ("COD\_CARTE")

REFERENCES "EDI"."CARTE" ("COD\_CARTE") ON DELETE CASCADE ENABLE;

ALTER TABLE "EDI"."REZERVA" ADD CONSTRAINT "FK\_REZERVA\_MEMBRU" FOREIGN KEY ("COD\_MEMBRU")

REFERENCES "EDI"."MEMBRU" ("COD\_MEMBRU") ON DELETE CASCADE ENABLE;



* 1. Tabela SCRIE

CREATE TABLE "EDI"."SCRIE"

( "COD\_CARTE" NUMBER(6,0),

"COD\_AUTOR" NUMBER(6,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.SCRIE

SET DEFINE OFF;

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (1,0);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (8,0);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (7,1);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (4,2);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (5,2);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (9,3);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (13,3);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (12,4);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (2,5);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (11,6);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (11,7);

Insert into EDI.SCRIE (COD\_CARTE,COD\_AUTOR) values (3,8);

--------------------------------------------------------

-- DDL for Index PK\_SCRIE\_AUTOR\_CARTE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_SCRIE\_AUTOR\_CARTE" ON "EDI"."SCRIE" ("COD\_AUTOR", "COD\_CARTE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table SCRIE

--------------------------------------------------------

ALTER TABLE "EDI"."SCRIE" ADD CONSTRAINT "PK\_SCRIE\_AUTOR\_CARTE" PRIMARY KEY ("COD\_AUTOR", "COD\_CARTE")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table SCRIE

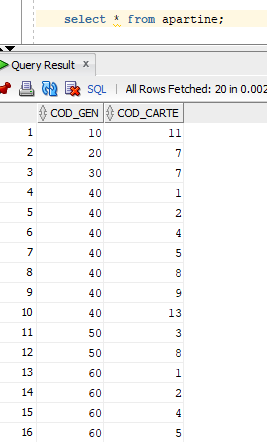
--------------------------------------------------------

ALTER TABLE "EDI"."SCRIE" ADD CONSTRAINT "FK\_SCRIE\_COD\_AUTOR" FOREIGN KEY ("COD\_AUTOR")

REFERENCES "EDI"."AUTOR" ("COD\_AUTOR") ON DELETE CASCADE ENABLE;

ALTER TABLE "EDI"."SCRIE" ADD CONSTRAINT "FK\_SCRIE\_COD\_CARTE" FOREIGN KEY ("COD\_CARTE")

REFERENCES "EDI"."CARTE" ("COD\_CARTE") ON DELETE CASCADE ENABLE;



* 1. Tabela APARTINE

CREATE TABLE "EDI"."APARTINE"

( "COD\_GEN" NUMBER(6,0),

"COD\_CARTE" NUMBER(6,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.APARTINE

SET DEFINE OFF;

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (10,11);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (20,7);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (30,7);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,1);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,2);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,4);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,5);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,8);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,9);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (40,13);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (50,3);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (50,8);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (60,1);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (60,2);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (60,4);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (60,5);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (60,9);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (60,13);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (70,12);

Insert into EDI.APARTINE (COD\_GEN,COD\_CARTE) values (100,3);

--------------------------------------------------------

-- DDL for Index PK\_APARTINE\_GEN\_CARTE

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_APARTINE\_GEN\_CARTE" ON "EDI"."APARTINE" ("COD\_GEN", "COD\_CARTE")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table APARTINE

--------------------------------------------------------

ALTER TABLE "EDI"."APARTINE" ADD CONSTRAINT "PK\_APARTINE\_GEN\_CARTE" PRIMARY KEY ("COD\_GEN", "COD\_CARTE")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table APARTINE

--------------------------------------------------------

ALTER TABLE "EDI"."APARTINE" ADD CONSTRAINT "FK\_APARTINE\_COD\_CARTE" FOREIGN KEY ("COD\_CARTE")

REFERENCES "EDI"."CARTE" ("COD\_CARTE") ON DELETE CASCADE ENABLE;

ALTER TABLE "EDI"."APARTINE" ADD CONSTRAINT "FK\_APARTINE\_COD\_GEN" FOREIGN KEY ("COD\_GEN")

REFERENCES "EDI"."GEN" ("COD\_GEN") ON DELETE CASCADE ENABLE;

* 1. Tabela CUMPARA

CREATE TABLE "EDI"."CUMPARA"

( "COD\_MEMBRU" NUMBER(6,0),

"COD\_ABONAMENT" NUMBER(6,0),

"COD\_PLATA\_ABONAMENT" NUMBER(6,0)

) SEGMENT CREATION IMMEDIATE

PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

REM INSERTING into EDI.CUMPARA

SET DEFINE OFF;

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (0,160,23);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (0,160,24);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (0,160,25);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (0,160,26);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (0,160,27);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (1,110,18);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (1,110,19);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (1,110,20);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (1,110,21);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (1,110,22);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (2,110,15);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (2,110,16);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (2,110,17);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (3,110,11);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (3,110,12);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (3,110,13);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (3,110,14);

Insert into EDI.CUMPARA (COD\_MEMBRU,COD\_ABONAMENT,COD\_PLATA\_ABONAMENT) values (4,140,10);

--------------------------------------------------------

-- DDL for Index PK\_CUMPARA

--------------------------------------------------------

CREATE UNIQUE INDEX "EDI"."PK\_CUMPARA" ON "EDI"."CUMPARA" ("COD\_MEMBRU", "COD\_ABONAMENT", "COD\_PLATA\_ABONAMENT")

PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ;

--------------------------------------------------------

-- Constraints for Table CUMPARA

--------------------------------------------------------

ALTER TABLE "EDI"."CUMPARA" ADD CONSTRAINT "PK\_CUMPARA" PRIMARY KEY ("COD\_MEMBRU", "COD\_ABONAMENT", "COD\_PLATA\_ABONAMENT")

USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS

STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645

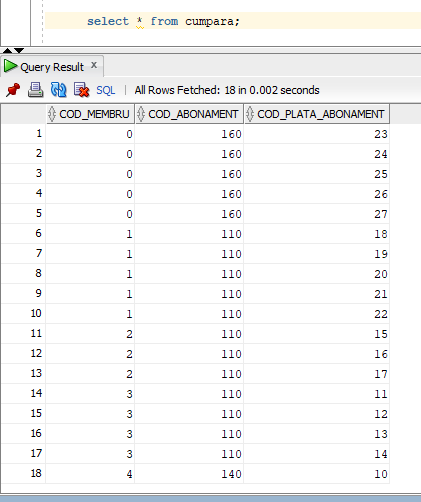
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER\_POOL DEFAULT FLASH\_CACHE DEFAULT CELL\_FLASH\_CACHE DEFAULT)

TABLESPACE "SYSTEM" ENABLE;

--------------------------------------------------------

-- Ref Constraints for Table CUMPARA

--------------------------------------------------------

 ALTER TABLE "EDI"."CUMPARA" ADD CONSTRAINT "FK\_CUMPARA\_ABONAMENT" FOREIGN KEY ("COD\_ABONAMENT")

REFERENCES "EDI"."ABONAMENT" ("COD\_ABONAMENT") ON DELETE CASCADE ENABLE;

ALTER TABLE "EDI"."CUMPARA" ADD CONSTRAINT "FK\_CUMPARA\_MEMBRU" FOREIGN KEY ("COD\_MEMBRU")

REFERENCES "EDI"."MEMBRU" ("COD\_MEMBRU") ON DELETE CASCADE ENABLE;

ALTER TABLE "EDI"."CUMPARA" ADD CONSTRAINT "FK\_CUMPARA\_PLATA\_ABONAMENT" FOREIGN KEY ("COD\_PLATA\_ABONAMENT")

REFERENCES "EDI"."PLATA\_ABONAMENT" ("COD\_PLATA\_ABONAMENT") ON DELETE CASCADE ENABLE;

1. Cererile SQL complexe

-- Sa se obtina codul, numele si prenumele membrilor ce au primit penalizari acum un an sau mai mult si au imprumutat cel putin 2 carti scrise in limba romana de un autor roman.

--Aceasta cerere utilizeaza urmatoarele:

--subcerere nesincronizata pe minim 3 tabele

--2 functii pe date (months\_between si sysdate)

--o functie pe siruri de caractere(lower)

--un bloc de cerere(clauza with)

--filtrare la nivel de linie

--grupare de date, functie de grup(count) si filtrare la nivel de grupuri

with mem\_2carti\_rom as

(

select m.cod\_membru

from carte c, membru m, imprumuta i

where c.cod\_carte = i.cod\_carte and m.cod\_membru = i.cod\_membru

and c.cod\_carte in (select c.cod\_carte

from carte c, scrie s, autor a

where c.cod\_carte = s.cod\_carte and a.cod\_autor = s.cod\_autor and lower(limba) = 'romana'

and lower(nationalitate) = 'roman')

group by m.cod\_membru

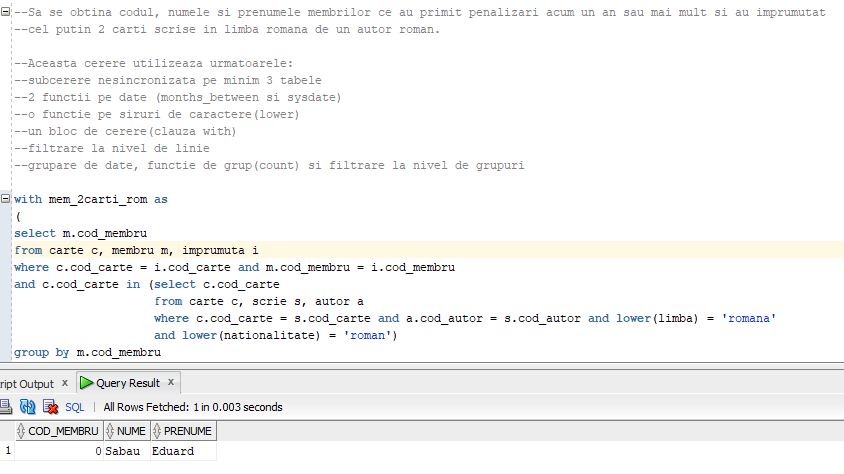
having count(c.cod\_carte) >= 2

)

select distinct m.cod\_membru, nume, prenume

from membru m, penalizare p, mem\_2carti\_rom m2

where m.cod\_membru = p.cod\_membru and months\_between(sysdate, p.data\_penalizare) >= 12 and m.cod\_membru = m2.cod\_membru;



--Se aplica mariri de preturi pentru abonamente din cauza inflatiei, cel de elev devine 20 de lei, cele de pensionar si de student cresc cu 20%, iar cele de adult cresc cu 30%. Afisati abonametele cu preturile modificate si ordonati rezultatele descrescator dupa pret.

--Aceasta cerere utilizeaza urmatoarele:

--o functie pe siruri de caractere(upper)

--expresia decode

--ordonare

select cod\_abonament, titlu, durata, decode(upper(titlu),

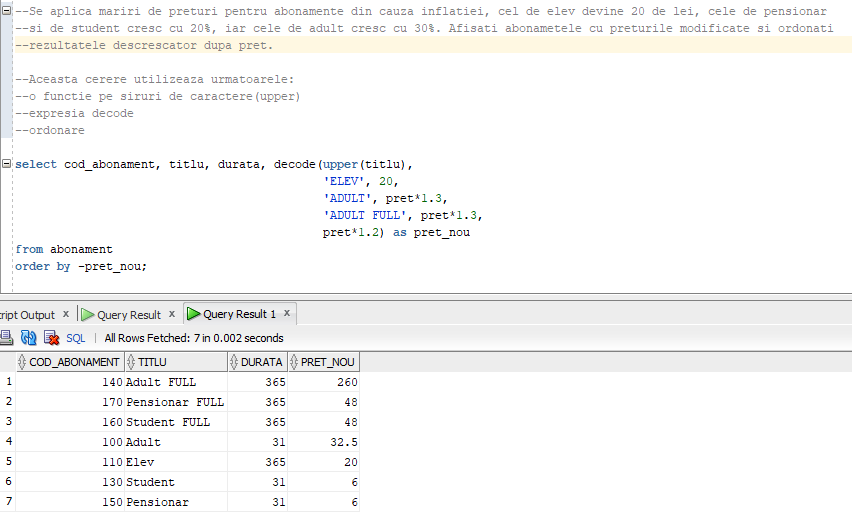
'ELEV', 20,

'ADULT', pret\*1.3,

'ADULT FULL', pret\*1.3,

pret\*1.2) as pret\_nou

from abonament

order by -pret\_nou;

--Sa se obtina codul, numele si prenumele membrilor ce au imprumutat o carte ce apartine genului 'Realism' imprumutata de cel putin un membru al carui nume incepe cu litera P si sa se ordoneze alfabetic dupa prenume rezultatele

-Aceasta cerere utilizeaza urmatoarele:

--o operatie de join pe cel putin 4 tabele

--subcerere sincronizata pe minim 3 tabele

--ordonare

--o functie pe sir de caractere (initcap)

select distinct m.cod\_membru, nume, prenume

from membru m, imprumuta i

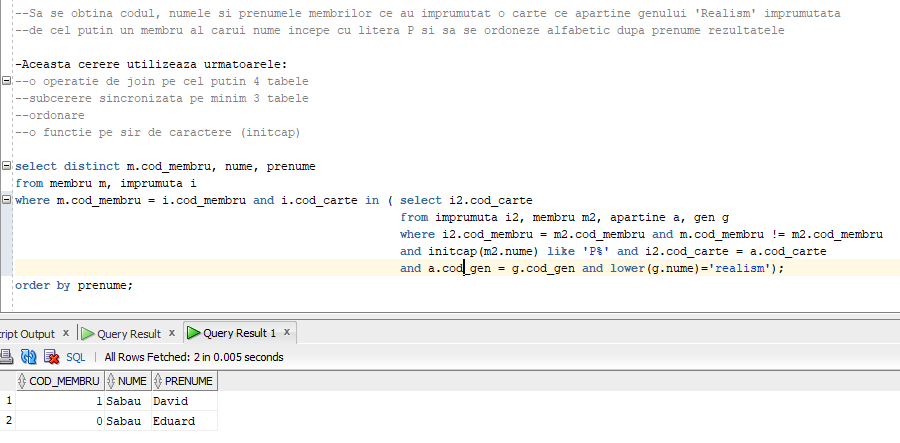
where m.cod\_membru = i.cod\_membru and i.cod\_carte in ( select i2.cod\_carte

from imprumuta i2, membru m2, apartine a, gen g

where i2.cod\_membru = m2.cod\_membru and m.cod\_membru != m2.cod\_membru

and initcap(m2.nume) like 'P%' and i2.cod\_carte = a.cod\_carte

and a.cod\_gen = g.cod\_gen and lower(g.nume)='realism');

order by prenume;

--Se pare ca inflatia a afectat si penalizarile neplatite. Penalizarile neplatite mai vechi de anul 2021 se scumpesc cu 50%, cele mai vechi de anul 2022 cu 25%, iar restul cu 15%. Ordonati crescator rezultatele dupa data in care a fost aplicata penalizarea.

--Aceasta cerere utilizeaza urmatoarele:

--instructiunea case

--functie pe date calendaristice(to\_date)

select cod\_penalizare, data\_penalizare, case

when data\_penalizare < to\_date('1/1/2021','DD/MM/YYYY') then 1.5\*suma\_penalizare

when data\_penalizare < to\_date('1/1/2022','DD/MM/YYYY') then 1.25\*suma\_penalizare

else 1.15\*suma\_penalizare

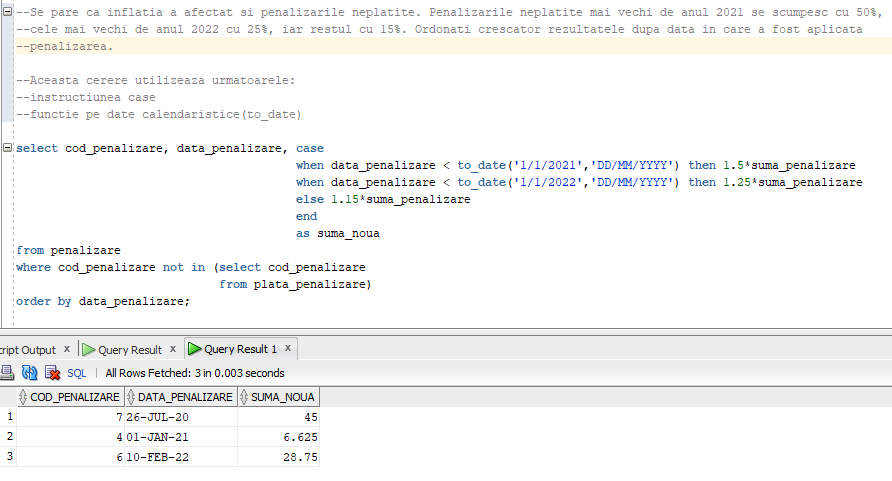
end

as suma\_noua

from penalizare

where cod\_penalizare not in (select cod\_penalizare

from plata\_penalizare)

order by data\_penalizare;

--Pentru fiecare carte din biblioteca, afisati codul, numarul de copii, isbn-ul si un mesaj corespunzator daca aceasta a fost sau nu imprumutata si daca a fost sau nu rezervata.

--Aceasta cerere utilizeaza urmatoarele:

--instructiunea nvl2

--2 subcereri corelate simple

select c.cod\_carte, titlu, numar\_copii, ISBN,

nvl2((select distinct cod\_carte

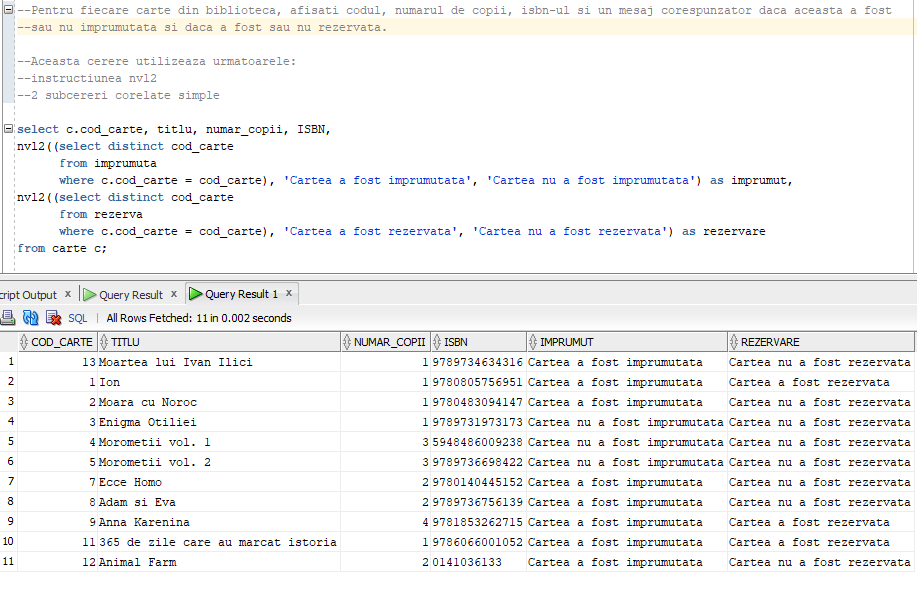
from imprumuta

where c.cod\_carte = cod\_carte), 'Cartea a fost imprumutata', 'Cartea nu a fost imprumutata') as imprumut,

nvl2((select distinct cod\_carte

from rezerva

where c.cod\_carte = cod\_carte), 'Cartea a fost rezervata', 'Cartea nu a fost rezervata') as rezervare

from carte c;

1. Cele 3 operații de actualizare și suprimare a datelor.

--Sa se stearga toate cartile care nu au fost imprumutate de membrul/membrii cu cele mai multe penalizari.

delete

from carte

where cod\_carte in (select cod\_carte

from imprumuta

where cod\_membru in (select m.cod\_membru

from penalizare p, membru m

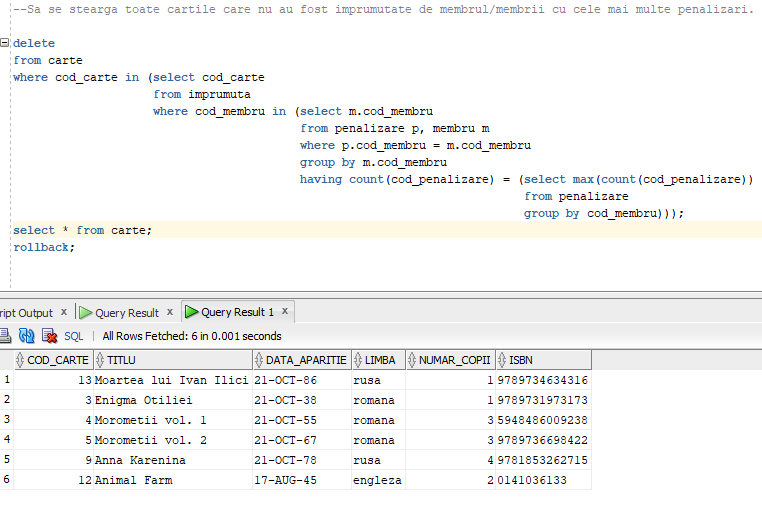
where p.cod\_membru = m.cod\_membru

group by m.cod\_membru

having count(cod\_penalizare) = (select max(count(cod\_penalizare))

from penalizare

group by cod\_membru)));

rollback;

--Biblioteca a hotarat ca toate abonamentele cumparate de un membru ce a imprumutat cartea cu titlul "Ion" la un moment dat sa devina gratis.

update abonament

set pret = 0

where cod\_abonament

in (select distinct a.cod\_abonament

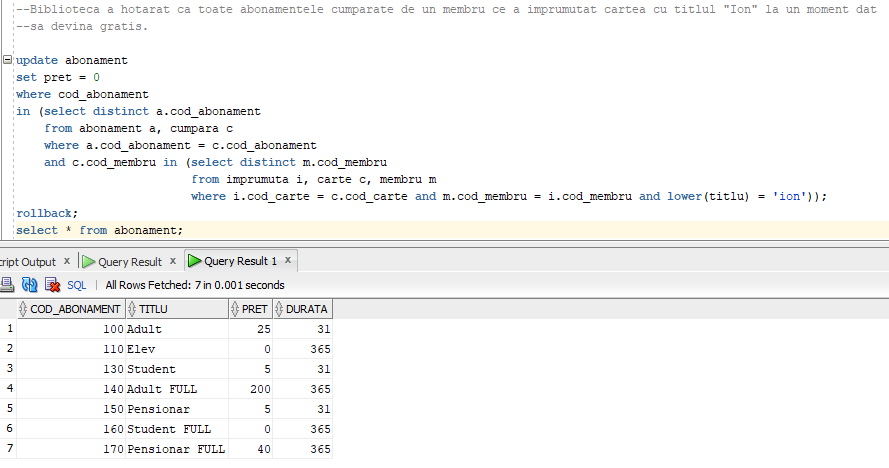
from abonament a, cumpara c

where a.cod\_abonament = c.cod\_abonament

and c.cod\_membru in (select distinct m.cod\_membru

from imprumuta i, carte c, membru m

where i.cod\_carte = c.cod\_carte and m.cod\_membru = i.cod\_membru and lower(titlu) = 'ion'));

rollback;

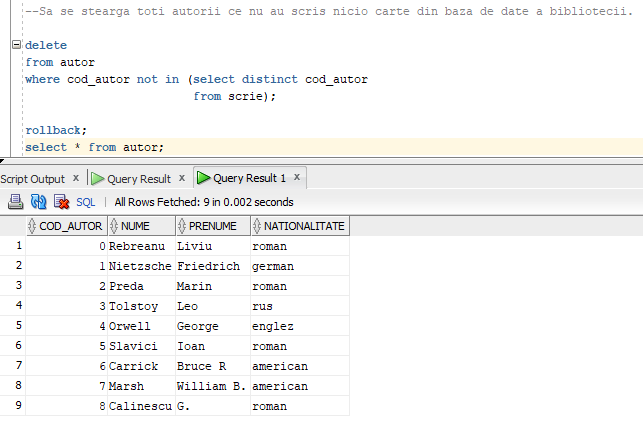
--Sa se stearga toti autorii ce nu au scris nicio carte din baza de date a bibliotecii.

delete

from autor

where cod\_autor not in (select distinct cod\_autor

from scrie);

rollback;

16. Outer-join și division.

--(outer-join) Sa se obtina pentru fiecare carte ce apartine unui gen ce contine 'ist' in nume de cate ori a fost imprumutata si sa se ordoneze descrescator rezultatele in functie de numarul de dati in care a fost imprumutata.

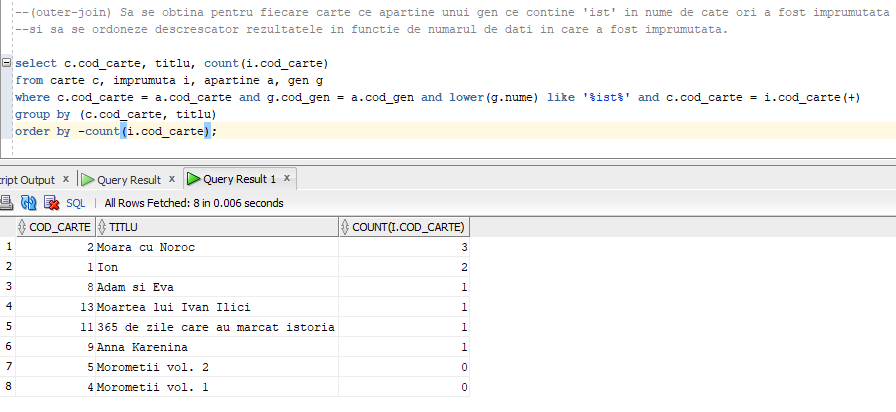
select c.cod\_carte, titlu, count(i.cod\_carte)

from carte c, imprumuta i, apartine a, gen g

where c.cod\_carte = a.cod\_carte and g.cod\_gen = a.cod\_gen and lower(g.nume) like '%ist%' and c.cod\_carte = i.cod\_carte(+)

group by (c.cod\_carte, titlu)

order by -count(i.cod\_carte);



--(division1) Sa se afiseze toate informatiile despre toti membri ce au imprumutat toate cartile scrise in limba rusa.

with mem\_carti\_rom as

(

select cod\_membru

from imprumuta

minus

select cod\_membru

from

(select cod\_membru, cod\_carte

from (select cod\_carte from carte where limba = 'rusa') t1,

(select cod\_membru from imprumuta) t2

minus

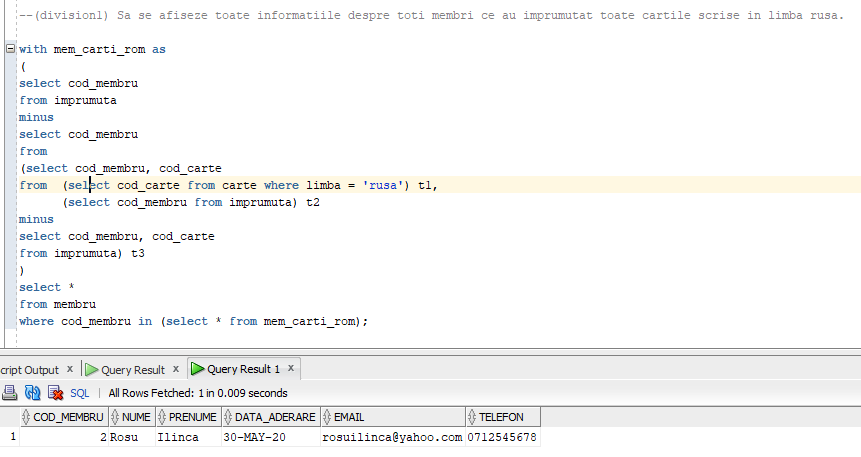
select cod\_membru, cod\_carte

from imprumuta) t3

)

select \*

from membru

where cod\_membru in (select \* from mem\_carti\_rom);

--(division2) Sa se afiseze toate informatiile despre toate abonamentele care au fost cumparate de toti membri ce au cel putin 3 penalizari

with ab\_cump\_mem as

(

select distinct cod\_abonament

from cumpara

where cod\_membru in (select cod\_membru

from penalizare

group by cod\_membru

having count(cod\_membru) >= 3)

group by cod\_abonament, cod\_plata\_abonament

having count(cod\_membru) = (select count(\*)

from (select cod\_membru

from penalizare

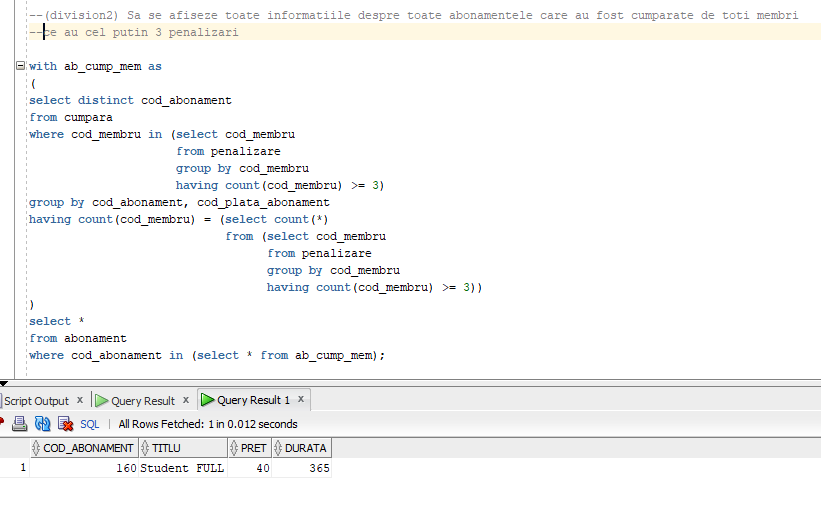
group by cod\_membru

having count(cod\_membru) >= 3))

)

select \*

from abonament

where cod\_abonament in (select \* from ab\_cump\_mem);

17. Optimizare cerere SQL

Să se obtină codul, numele, prenumele membrilor și codurile, sumele și datele penalizărilor pentru care data penalizării este după 13.06.2021, dar înainte de 20.04.2022.

Varianta neoptimizată:

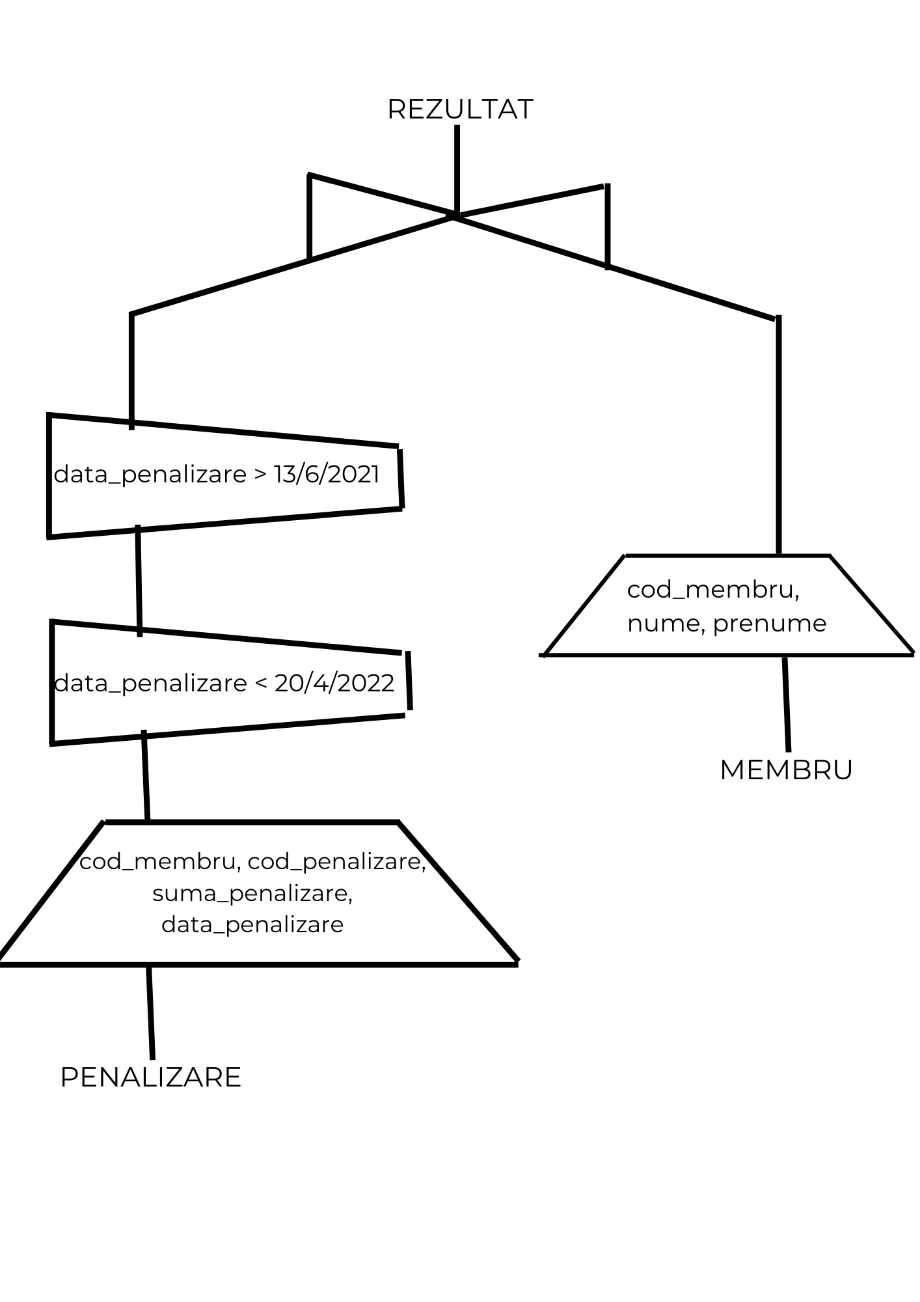
R1 = PROJECT(PENALIZARE, cod\_membru, cod\_pennalizare, suma\_penalizare, data\_penalizare)

R2 = SELECT(R1, data\_penalizare > 13/6/2021)

R3 = SELECT(R2, data\_penalizare < 20/4/2022)

R4 = PROJECT(MEMBRU, cod\_membru, nume, prenume)

Rezultat=JOIN(R4, R3)



--Sa se obtina codul, numele, prenumele membrilor si codurile, sumele si datele penalizarilor pentru care data penalizarii

--este dupa 13.06.2021, dar inainte de 20.04.2022

with pens as

(

select cod\_membru, cod\_penalizare, suma\_penalizare, data\_penalizare

from (select cod\_membru, cod\_penalizare, suma\_penalizare, data\_penalizare

from penalizare

where data\_penalizare > to\_date('13/6/2021', 'DD/MM/YYYY'))

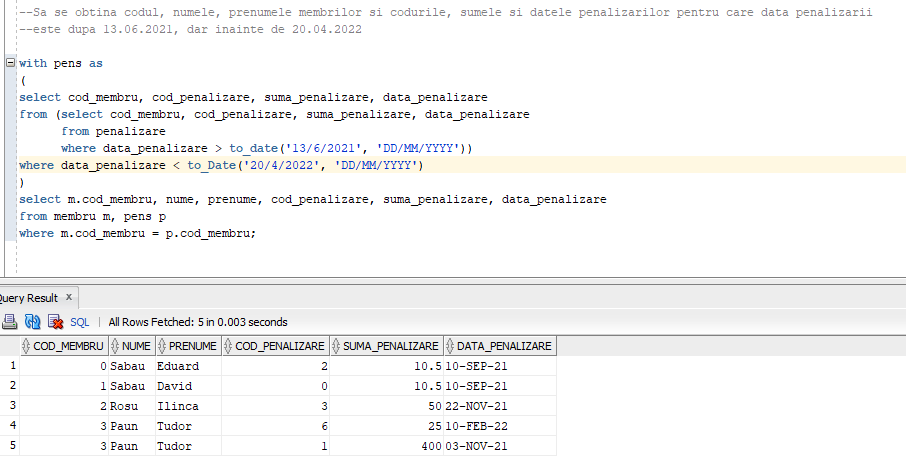
where data\_penalizare < to\_Date('20/4/2022', 'DD/MM/YYYY')

)

select m.cod\_membru, nume, prenume, cod\_penalizare, suma\_penalizare, data\_penalizare

from membru m, pens p

where m.cod\_membru = p.cod\_membru;

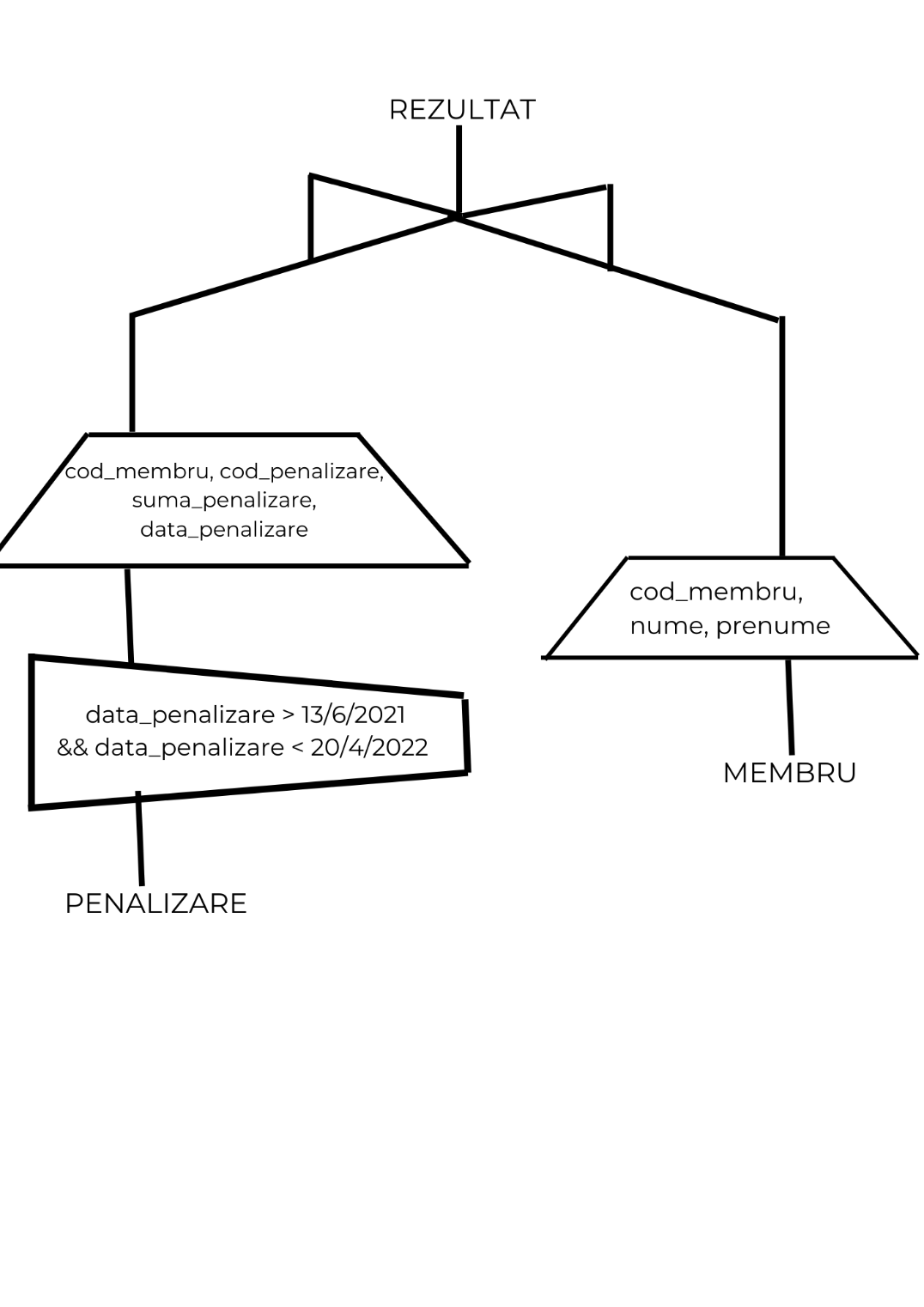


Varianta optimizată:

R1 = PROJECT(SELECT(PENALIZARE, data\_penalizare > 13/6/2020 && data\_penalizare < 20/4/2022), cod\_membru, cod\_penalizare, suma\_penalizare, data\_penalizare)

R2 = PROJECT(MEMRU, cod\_membru, nume, prenume)

REZULTAT = JOIN(R1, R2)



--Sa se obtina codul, numele, prenumele membrilor si codurile, sumele si datele penalizarilor pentru care data penalizarii

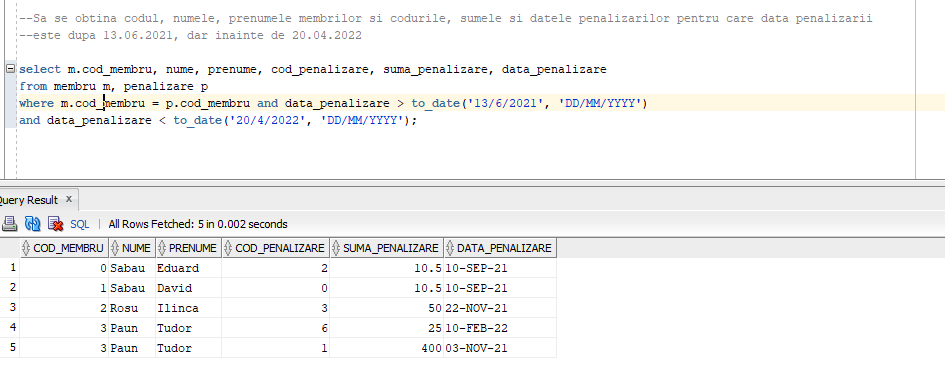
--este dupa 13.06.2021, dar inainte de 20.04.2022

select m.cod\_membru, nume, prenume, cod\_penalizare, suma\_penalizare, data\_penalizare

from membru m, penalizare p

where m.cod\_membru = p.cod\_membru and data\_penalizare > to\_date('13/6/2021', 'DD/MM/YYYY')

and data\_penalizare < to\_date('20/4/2022', 'DD/MM/YYYY');



18. Formele normale superioare

18.1 Forma normală Boyce-Codd

Presupunem că în tabela MEMBRU am avea atributul adresa ca parte din cheie primară și atributul cod\_postal ca atribut normal cu următoarea schemă relațională

MEMBRU(cod\_membru#, adresa#, nume, prenume, data\_aderare, email, telefon, cod\_postal)

În acest caz atributul adresa care face parte din cheia primară depinde de cod\_postal care este un atribut simplu. Soluția ar fi ca atributele să fie inversate, cod\_postal devenind parte din cheia primară și adresa atribut normal. Pentru tabela din diagrama mea conceptuală am ales să șterg cu totul cele 2 atribute întrucât nu au relevanță pentru modelul proiectat.