Proiect baze de date
Organizarea unui festival

Baidoc Robert-Cristian
Grupa 244

Cuprins

1. Descrierea modelului real, a utilitatii acestuia si a regulilor de functionare

Tema acestui proiect este o baza de date pentru gestionarea festivalelor si pentru retinerea de informatii despre aceste.

Aceasta baza de date poate fi utilizata pentru a organiza eficient festivalurile si pentru a monitoriza tot ce se intampla in cadrul acestora.

Entitatea “FESTIVALURI” contine date despre festival, cum ar fi numele, data desfasurarii acestuia si localitatea. Aceasta este legata de mai multe entitati: “BILETE”, “SCENE” si “ANGAJATI”. Entitatile “ STANDARD”, “VIP” si “GOLD” contin dat despre fiecare tip de bilet pus la vanzare. De asemenea, fiecare bilet contine si date despre clientul care l-a achizitionat. Datele despre clienti se afla in tabelul “CLIENTI”. Entitatea “SCENE” este legata de entitatea “CONCERTE”, deoarece fiecare concert este realizat pe o scena. Intre entitatile “ANGAJATI” si “CONCERTE” exista o relatie many to many, deoarece un anajat poate lucra la mai multe concerte, iar la un concert pot lucra mai multi angajati. De aceea este prezenta tabela “ANGAJATI_CONCERTE”. De entitatea “CONCERTE” mai este legata si entitatea “ARTISTI”, deoarece fiecare concert este sustinut de un artist.

Utilitatea acestei baze de date este de a face gestionarea unui festival mai usoara, avand la indemana date despre majoritatea entitatilor importante din organizarea unui festival.

Functionalitatile acestei baze de date includ stocarea de date despre fiecare element descris mai sus(ex: bilete, scene, concerte) si accesul rapid la informatii despre acestea.

2. Prezentarea constrangerilor(restrictii, reguli) impuse asupra modelului

1. Tabela FESTIVALURI
    -id_festival este cheia primara si trebuie sa fie unica pentru fiecare inregistrare
    -nume_festival trebuie sa aiba o lungime maxima de 50 de caractere
    -locatie trebuie sa aiba o lungime maxima de 100 de caractere
    -data_inceput si data_final trebuie sa fie de tip DATE, iar data_final sa fie mai mare decat data_inceput

2. Tabela CLIENTI
    -id_client este cheia primara si trebuie sa fie unica pentru fiecare inregistrare
    -nume si prenume trebuie sa aiba o lungime maxima de 30 de caractere
    -data_nastere trebuie sa fie de tip DATE
    -email trebuie sa fie un sir de caractere
    -nr_telefon trebuie sa fie un sir de caractere de 10 cifre

3. Tabela BILETE

-id_bilet este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_client este cheie externa catre client si trebuie sa exista in acea tabela

-id_festival este cheie externa catre festival si trebuie sa existe in acea tabela

-data_achizitionare este de tip DATE

4. Tabela STANDARD

-id_standard este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_bilet este cheie externa(si parte din PK) si trebuie sa existe in tabela bilete

-pret trebuie sa fie o valoare pozitiva

5. Tabela VIP

-id_vip este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_bilet este cheie externa(si parte din PK) si trebuie sa existe in tabela bilete

-pret trebuie sa fie o valoare pozitiva

-nr_bauturi trebuie sa fie o valoare pozitiva

6. Tabela GOLD

-id_gold este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_bilet este cheie externa(si parte din PK) si trebuie sa existe in tabela bilete

-pret trebuie sa fie o valoare pozitiva

-nr_bauturi trebuie sa fie o valoare pozitiva

-extra trebuie sa aiba o lungime de maxim 100 de caractere

7. Tabela ARTISTI

-id_artist este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-nume trebuie sa aiba o lungime de maxim 50 de caractere

-email_agent trebuie sa aiba o lungime de maxim 50 de caractere

-nationalitate trebuie sa aiba o lungime de maxim 30 de caractere

-pret_concert trebuie sa fie o valoare pozitiva

8. Tabela CONCERTE

-id_concert este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_artist si id_scena sunt chei externe si trebuie sa existe in ARTISTI si SCENE

-data este de tip DATE

-ora_inceput si ora_final au o lungime de maxim 8 caractere

9. Tabela SCENE

-id_scena este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_festival e cheie externe si trebuie sa existe in FESTIVALURI

-nume are lungime maxima de 30 de caractere

-sponsor are lungime maxima de 50 de caractere

-nr_scena trebuie sa fie un numar pozitiv

10. Tabela ANGAJATI

-id_angajat este cheia primara si trebuie sa fie unica pentru fiecare inregistrare

-id_festival este cheie externa si trebuie sa existe in FESTIVALURI

-rol are lungime maxima de 30 de caractere

-salariu este o valoare pozitiva

11. Tabela ANGAJATI_CONCERTE

-are doua coloane cheie care fac referire la cheile primare ale tabelelor “ANGAJATI” si “CONCERTE” . Acestea doua formeaza impreuna cheia primara a tabelului

3. Descrierea entitatilor, incluzand precizarea cheii primare

1. Entitatea FESTIVALURI

Aceasta tabela contine informatii despre festivalurile de muzica organizate. Fiecare inregistrare este identificata prin cheia primara id_festival si contine informatii despre festival(nume_festival), locatia acestuia, precum si datele de inceput si de final.

2. Entitatea CLIENTI

Aceasta tabela contine informatii despre persoanele care achizitioneaza bilete Festivaluri. Fiecare client este identificat prin cheia prima id_client si are asociate infromatii precum nume, prenume, data nastere, email si numar de telefon.

3. Entitatea BILETE

Aceasta tabela contine informatii despre biletele cumparate de catre clienti pentru festivaluri. Cheia primara este id_bilet. Sunt inregistrate si chei externe catre id_client si id_festival. Este inregistrata si data de achizitionare a biletului.

4. Entitatea SCENE

Aceasta tabela contine informatii despre scenele disponibile in cadrul fiecarui festival. Fiecare scena este identificata prin id_scena, care reprezinta cheia primara, si este legata de un festival prin id_festival (cheie externa). Se mai retin informatii precum nume, sponsor si nr_scena, care permit diferentierea si organizarea scenelor in cadrul evenimentului.

5. Entitatea ARTISTI

Aceasta tabela contine informatii despre artistii care participa la festivaluri. Fiecare artist este identificat prin id_artist, cheia primara, si este asociat cu date precum nume, email_agent, nationalitate si pret_concert. Aceste date sunt importante pentru gestionarea contractelor si programarii concertelor.

6. Entitatea CONCERTE

Aceasta tabela stocheaza informatii despre concertele care au loc in cadrul festivalurilor. Cheia primara este id_concert, iar tabela contine chei externe catre id_artist si id_scena, pentru a indica cine canta si unde. Se pastreaza si informatii legate de data, ora_inceput si ora_final. Fiecare concert este astfel bine localizat in timp si spatiu.

7. Entitatea ANGAJATI

Aceasta tabela contine informatii despre angajatii festivalurilor. Fiecare angajat este identificat prin id_angajat, cheia primara, si are asociate atribute precum rol, salariu si id_festival (cheie externa care arata la ce festival lucreaza). Tabela permite gestionarea resurselor umane in cadrul fiecarui festival.

8. Entitatea ANGAJATI_CONCERTE

Aceasta tabela este una asociativa, care leaga angajatii de concertele la care participa in mod direct. Cheia primara este compusa din id_angajat si id_concert, iar ambele sunt chei externe catre entitatile corespunzatoare. Tabela reflecta implicarea concreta a personalului in organizarea fiecarui concert.

9. Entitatea STANDARD / VIP / GOLD

Aceste trei tabele sunt subentitati ale entitatii BILETE si descriu tipul biletului achizitionat. Fiecare tabel are o cheie primara proprie (id_standard, id_vip, id_gold), dar si o cheie externa comuna id_bilet. Se retin atribute specifice: pret pentru toate tipurile, nr_bauturi pentru VIP si GOLD, respectiv extra pentru GOLD. Aceste tabele permit diferentierea ofertelor disponibile pentru clienti in functie de tipul biletului.

4.Descrierea relatiilor, incluzand precizarea cardinalitatii acestora

Tabela FESTIVALURI are o relație de tip „one-to-many” cu tabela SCENE, deoarece un festival poate avea mai multe scene (obligatoriu cel puțin una), însă fiecare scenă aparține unui singur festival. Cardinalitatea este 1:M(1).

Tabela FESTIVALURI are o relație de tip „one-to-many” cu tabela BILETE, deoarece un festival poate avea mai multe bilete asociate, dar fiecare bilet este emis pentru un singur festival. Cardinalitatea este 1:M(0).

Tabela CLIENTI are o relație de tip „one-to-one” cu tabela BILETE, deoarece un client poate cumpara un singur bilet (biletele sunt nominale) iar un bilet poate fi cumparat de un singur client. Cardinalitatea este 1:1.

Tabela SCENE are o relație de tip „one-to-many” cu tabela CONCERTE, deoarece pe o scenă se pot desfășura mai multe concerte (sau niciunul), dar fiecare concert are loc pe o singură scenă. Cardinalitatea este 1:M(0).

Tabela ARTISTI are o relație de tip „one-to-many” cu tabela CONCERTE, deoarece un artist poate susține mai multe concerte, dar fiecare concert este susținut de un singur artist. Cardinalitatea este 1:M(1).

Tabela CONCERTE are o relație de tip „many-to-many” cu tabela ANGAJATI, deoarece un concert poate fi organizat de mai mulți angajați, iar un angajat poate lucra la mai multe concerte. Această relație este implementată prin tabela asociativă ANGAJATI_CONCERTE, care are cardinalitatea M:M.

Tabela BILETE are o relație de tip „ISA” cu tabelele STANDARD, VIP și GOLD, deoarece fiecare bilet este de un singur tip (Standard, VIP sau Gold), iar toate aceste tipuri moștenesc atributele din tabela BILETE. Este o relație de tip „one-to-one”, adică 1:1, deoarece un bilet poate fi de un singur tip.

5.Descrierea atributelor, incluzand tipul de date eventualele constrangeri, valori implicite, valori posibile ale atributelor.

1. Tabela FESTIVALURI:

- **id_festival** (INT, PK): cheie primara a tabelului “FESTIVALURI”, stocata sub forma de numar intreg, generata automat prin secventa FESTIVALURI_SEQ.

- **nume_festival** (VARCHAR2(50)): reprezinta denumirea festivalului, stocata sub forma de sir de caractere cu lungimea maxima de 50.

- **locatie** (VARCHAR2(100)): indica locatia de desfasurare a festivalului, sub forma de sir de caractere cu lungimea maxima de 100.

- **data_inceput** (DATE): reprezinta data de inceput a festivalului.

- **data_final** (DATE): reprezinta data de incheiere a festivalului.

	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_FESTIVAL	NUMBER	22	N	"C##ROBERT"."FESTIVALURI_SEQ"."NEXTVAL"
2	NUME_FESTIVAL	VARCHAR2	50	Y	<null>
3	LOCATIE	VARCHAR2	100	Y	<null>
4	DATA_INCEPUT	DATE	7	Y	<null>
5	DATA_FINAL	DATE	7	Y	<null>

2. Tabela CLIENTI:

- **id_client** (INT, PK): cheie primara a tabelului “CLIENTI”, stocata sub forma de numar intreg, generata automat prin secventa CLIENTI_SEQ.

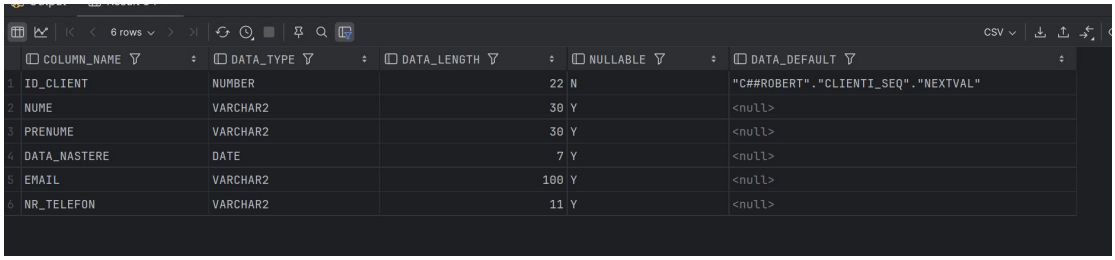
- **nume** (VARCHAR2(30)): reprezinta numele clientului, stocat sub forma de sir de caractere cu lungimea maxima de 30.

- **prenume** (VARCHAR2(30)): reprezinta prenumele clientului, stocat sub forma de sir de caractere cu lungimea maxima de 30.

- **data_nastere** (DATE): indica data nasterii clientului.

- **email** (VARCHAR2(100)): adresa de email a clientului, stocata sub forma de sir de caractere cu lungimea maxima de 100.

- **nr_telefon** (VARCHAR2(11)): numarul de telefon al clientului, stocat sub forma de sir de caractere cu lungimea maxima de 11.



6 rows — CSV

	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_CLIENT	NUMBER	22	N	"C##ROBERT"."CLIENTI_SEQ"."NEXTVAL"
2	NUME	VARCHAR2	30	Y	<null>
3	PRENUME	VARCHAR2	30	Y	<null>
4	DATA_NASTERE	DATE	7	Y	<null>
5	EMAIL	VARCHAR2	100	Y	<null>
6	NR_TELEFON	VARCHAR2	11	Y	<null>

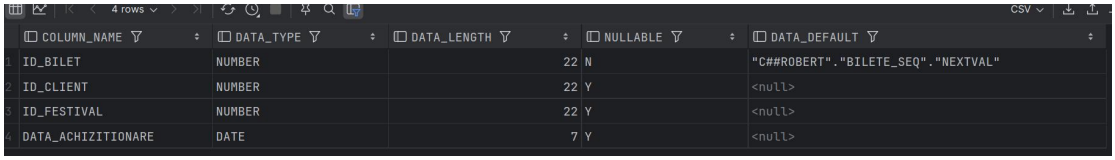
3. Tabela BILETE:

- **id_bilet** (INT, PK): cheie primara a tabelului “BILETE”, stocata sub forma de numar intreg, generata automat prin secventa BILETE_SEQ.

- **id_client** (INT): reprezinta identificatorul clientului care a achizitionat biletul. Este o cheie externa catre tabela CLIENTI.

- **id_festival** (INT): reprezinta identificatorul festivalului pentru care a fost emis biletul. Este o cheie externa catre tabela FESTIVALURI.

- **data_achizitionare** (DATE): indica data la care biletul a fost achizitionat.



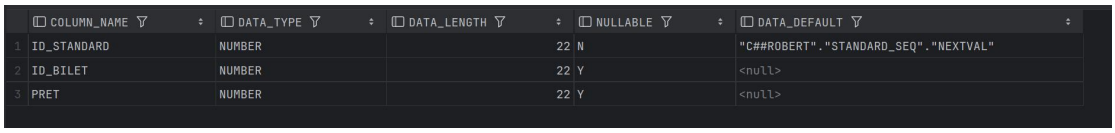
	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_BILET	NUMBER	22	N	"C##ROBERT"."BILETE_SEQ"."NEXTVAL"
2	ID_CLIENT	NUMBER	22	Y	<null>
3	ID_FESTIVAL	NUMBER	22	Y	<null>
4	DATA_ACHIZITIONARE	DATE	7	Y	<null>

4. Tabela STANDARD:

- **id_standard** (INT, PK): cheie primara a tabelului “STANDARD”, stocata sub forma de numar intreg, generata automat prin secventa STANDARD_SEQ.

- **id_bilet** (INT): reprezinta identificatorul biletului de tip standard. Este o cheie externa catre tabela BILETE.

- **pret** (NUMBER): reprezinta pretul biletului standard, stocat sub forma de numar.



	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_STANDARD	NUMBER	22	N	"C##ROBERT"."STANDARD_SEQ"."NEXTVAL"
2	ID_BILET	NUMBER	22	Y	<null>
3	PRET	NUMBER	22	Y	<null>

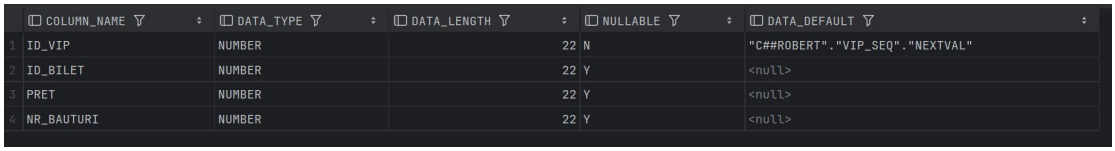
5. Tabela VIP:

- **id_vip** (INT, PK): cheie primara a tabelului “VIP”, stocata sub forma de numar intreg, generata automat prin secventa VIP_SEQ.

- **id_bilet** (INT): reprezinta identificatorul biletului de tip VIP. Este o cheie externa catre tabela BILETE.

- **pret** (NUMBER): reprezinta pretul biletului VIP, stocat sub forma de numar.

- **nr_bauturi** (NUMBER): indica numarul de bauturi incluse in biletul VIP.



	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_VIP	NUMBER	22	N	"C##ROBERT"."VIP_SEQ"."NEXTVAL"
2	ID_BILET	NUMBER	22	Y	<null>
3	PRET	NUMBER	22	Y	<null>
4	NR_BAUTURI	NUMBER	22	Y	<null>

6. Tabela GOLD:

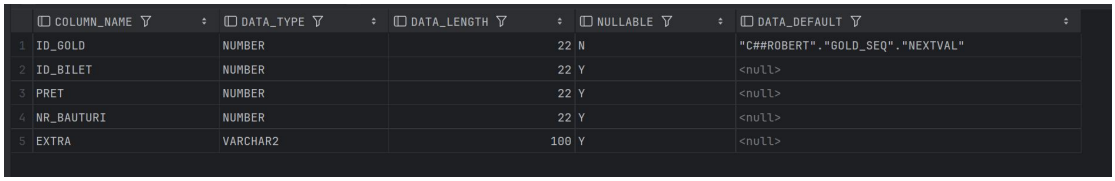
- **id_gold** (INT, PK): cheie primara a tabelului “GOLD”, stocata sub forma de numar intreg, generata automat prin secventa GOLD_SEQ.

- **id_bilet** (INT): reprezinta identificatorul biletului de tip GOLD. Este o cheie externa catre tabela BILETE.

- **pret** (NUMBER): reprezinta pretul biletului GOLD, stocat sub forma de numar.

- **nr_bauturi** (NUMBER): indica numarul de bauturi incluse in biletul GOLD.

- **extra** (VARCHAR2(100)): reprezinta beneficii suplimentare, cum ar fi acces backstage sau cadouri, stocate sub forma de sir de caractere cu lungimea maxima de 100.



	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_GOLD	NUMBER	22	N	"C##ROBERT"."GOLD_SEQ"."NEXTVAL"
2	ID_BILET	NUMBER	22	Y	<null>
3	PRET	NUMBER	22	Y	<null>
4	NR_BAUTURI	NUMBER	22	Y	<null>
5	EXTRA	VARCHAR2	100	Y	<null>

7. Tabela ARTISTI:

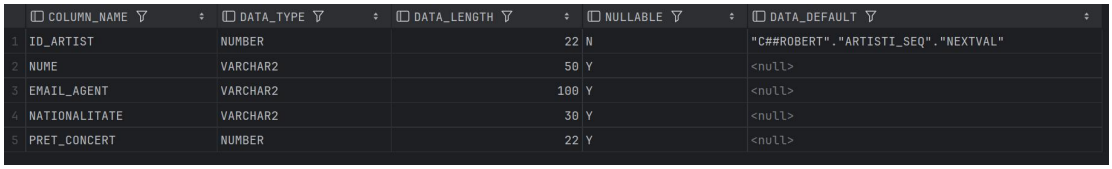
- **id_artist** (INT, PK): cheie primara a tabelului “ARTISTI”, stocata sub forma de numar intreg, generata automat prin secventa ARTISTI_SEQ.

- **nume** (VARCHAR2(50)): reprezinta numele artistului, stocat sub forma de sir de caractere cu lungimea maxima de 50.

- **email_agent** (VARCHAR2(100)): adresa de email a agentului artistului, stocata sub forma de sir de caractere cu lungimea maxima de 100.

- **nationalitate** (VARCHAR2(30)): indica nationalitatea artistului, stocata sub forma de sir de caractere cu lungimea maxima de 30.

- **pret_concert** (NUMBER): reprezinta suma perceputa de artist pentru sustinerea unui concert.



	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_ARTIST	NUMBER	22	N	"C##ROBERT"."ARTISTI_SEQ"."NEXTVAL"
2	NUME	VARCHAR2	50	Y	<null>
3	EMAIL_AGENT	VARCHAR2	100	Y	<null>
4	NATIONALITATE	VARCHAR2	30	Y	<null>
5	PRET_CONCERT	NUMBER	22	Y	<null>

8. Tabela CONCERTE:

- **id_concert** (INT, PK): cheie primara a tabelului “CONCERTE”, stocata sub forma de numar intreg, generata automat prin secventa CONCERTE_SEQ.

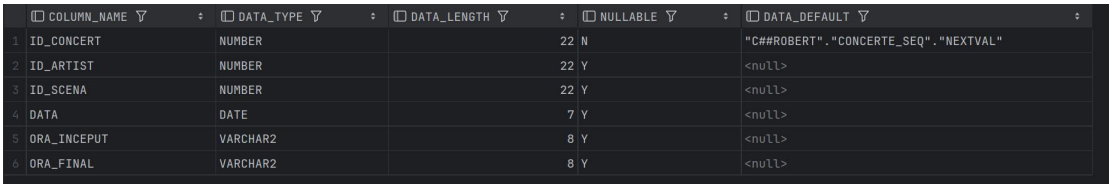
- **id_artist** (INT): reprezinta identificatorul artistului care sustine concertul. Este o cheie externa catre tabela ARTISTI.

- **id_scena** (INT): reprezinta identificatorul scenei pe care are loc concertul. Este o cheie externa catre tabela SCENE.

- **data** (DATE): reprezinta data la care are loc concertul.

- **ora_inceput** (VARCHAR2(8)): reprezinta ora de incepere a concertului, stocata sub forma de sir de caractere.

- **ora_final** (VARCHAR2(8)): reprezinta ora de incheiere a concertului, stocata sub forma de sir de caractere.



	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_CONCERT	NUMBER	22	N	"C##ROBERT"."CONCERTE_SEQ"."NEXTVAL"
2	ID_ARTIST	NUMBER	22	Y	<null>
3	ID_SCENA	NUMBER	22	Y	<null>
4	DATA	DATE	7	Y	<null>
5	ORA_INCEPUT	VARCHAR2	8	Y	<null>
6	ORA_FINAL	VARCHAR2	8	Y	<null>

9. Tabela SCENE:

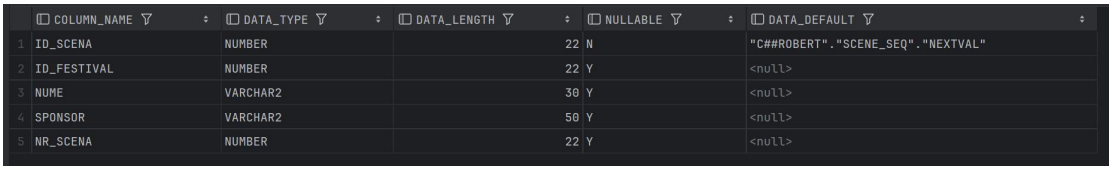
- **id_scena** (INT, PK): cheie primara a tabelului “SCENE”, stocata sub forma de numar intreg, generata automat prin secventa SCENE_SEQ.

- **id_festival** (INT): reprezinta identificatorul festivalului caruia ii apartine scena. Este o cheie externa catre tabela FESTIVALURI.

- **nume** (VARCHAR2(30)): reprezinta denumirea scenei, stocata sub forma de sir de caractere cu lungimea maxima de 30.

- **sponsor** (VARCHAR2(50)): numele sponsorului asociat scenei, stocat sub forma de sir de caractere cu lungimea maxima de 50.

- **nr_scena** (NUMBER): numarul de identificare al scenei in cadrul festivalului.



	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_SCENA	NUMBER	22	N	"C##ROBERT"."SCENE_SEQ"."NEXTVAL"
2	ID_FESTIVAL	NUMBER	22	Y	<null>
3	NUME	VARCHAR2	30	Y	<null>
4	SPONSOR	VARCHAR2	50	Y	<null>
5	NR_SCENA	NUMBER	22	Y	<null>

10. Tabela ANGAJATI:

- **id_angajat** (INT, PK): cheie primara a tabelului “ANGAJATI”, stocata sub forma de numar intreg.

- **id_festival** (INT): reprezinta identificatorul festivalului la care este alocat angajatul. Este o cheie externa catre tabela FESTIVALURI.

- **rol** (VARCHAR2(30)): reprezinta rolul angajatului in cadrul festivalului (ex: sunetist, lumini, logistica), stocat sub forma de sir de caractere cu lungimea maxima de 30.

- **salariu** (NUMBER): reprezinta salariul angajatului, stocat sub forma de numar.

	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_ANGAJAT	NUMBER	22	N	<null>
2	ID_FESTIVAL	NUMBER	22	Y	<null>
3	ROL	VARCHAR2	30	Y	<null>
4	SALARIU	NUMBER	22	Y	<null>

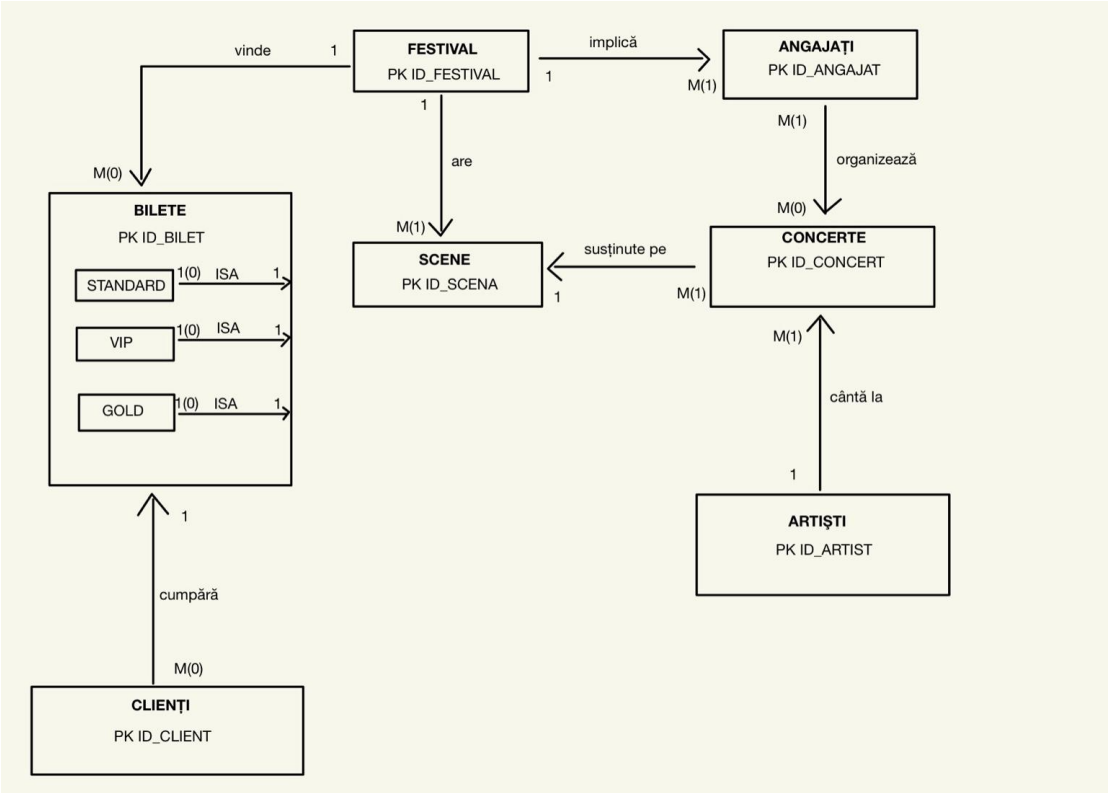
11. Tabela ANGAJATI_CONCERTE:

- **id_angajat** (INT, PK, FK): reprezinta identificatorul angajatului care participa la un concert. Este parte din cheia primara compusa si este cheie externa catre tabela ANGAJATI.

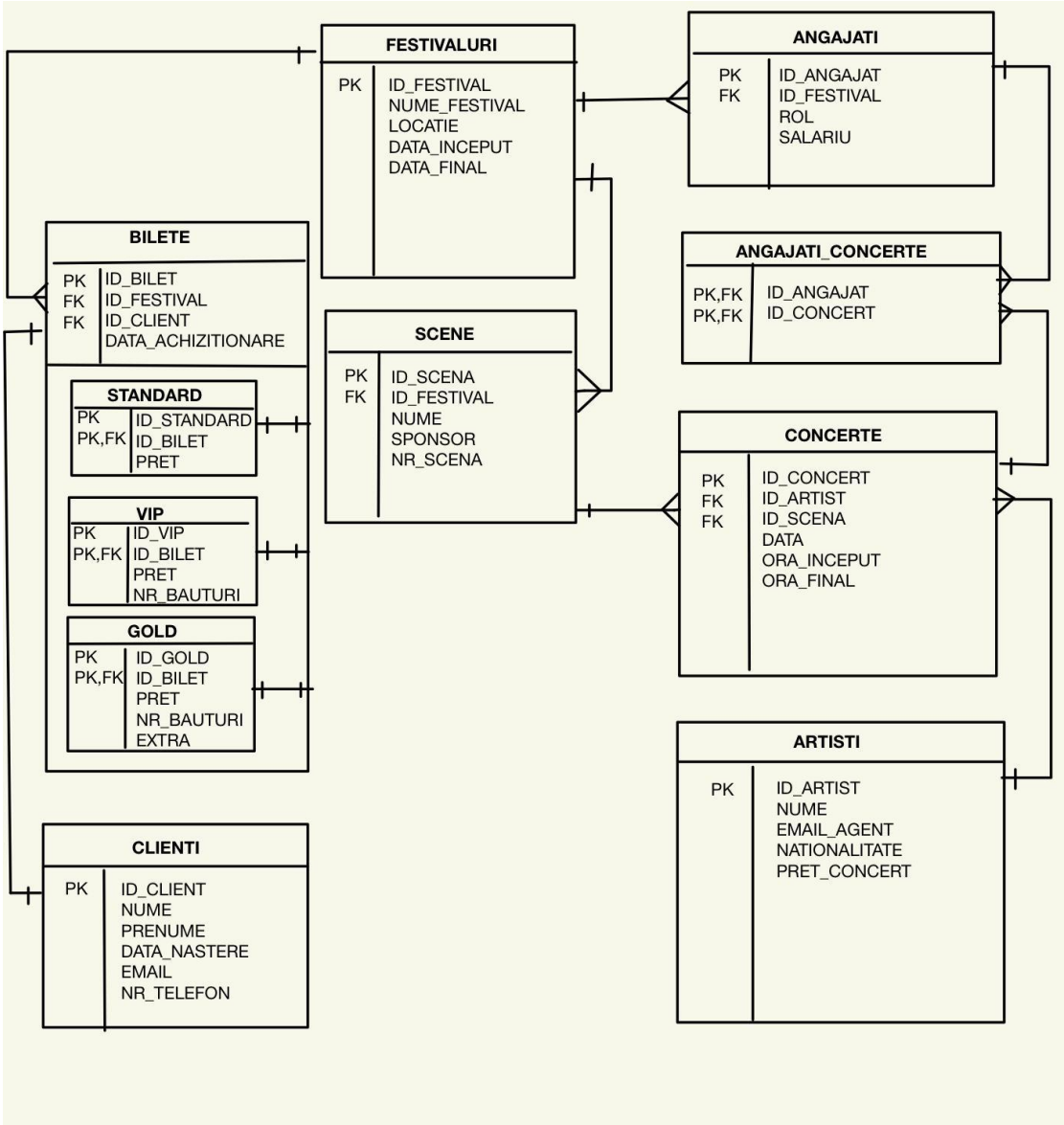
- **id_concert** (INT, PK, FK): reprezinta identificatorul concertului la care participa angajatul. Este parte din cheia primara compusa si este cheie externa catre tabela CONCERTE.

	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	DATA_DEFAULT
1	ID_ANGAJAT	NUMBER	22	N	<null>
2	ID_CONCERT	NUMBER	22	N	<null>

6.Realizarea diagramei entitate-relatie corespunzatoare descrierii de la punctele 3-5.



7.Realizarea diagramei conceptuale corespunzatoare corespunzatoare diagramei entitatie-relatie.



8.Enumerarea schemelor relationale corespunzatoare diagramei conceptuale

Schemele relationale:

-FESTIVALURI(id_festival, nume_festival, locatie, data_inceput, data_final)
-CLIENTI(id_client, nume, prenume, data_nastere, email, nr_telefon)
-BILETE(id_bilet, id_festival, id_client, data_achizitionare)
-STANDARD(id_standard, id_bilet, pret)

-VIP(id_vip, id_bilet, pret, nr_bauturi)
-GOLD(id_gold, id_bilet, pret, nr_bauturi, extra)
-ARTISTI(id_artist, nume, email_agent, nationalitate, pret_concert)
-SCENE(id_scena,id_festival, nume, sponsor, nr_scena)
-CONCERTE(id_concert,id_artist, id_scena, data, ora_inceput, ora_final)
-ANGAJATI(id_angajat, id_festival, rol, salariu)
-ANGAJATI_CONCERTE(id_angajat, id_concert)

9.Realizarea normalizarii pana la forma normala 3 (FN1-FN3)

FORMA NORMALA 1:

-O relatie se afla in forma normala 1 daca fiecarui atribut care o compune ii corespunde o valoare indivizibila. Mai precis, o relatie se afla în forma normala 1 daca exista un identificator unic.

-Exemplu, in entitatea ARTISTI exista cheia primara id_artist, identificator unic pentru fiecare artist. Id_angajat este o valoarea indivizabila, deci relatia se afla in forma normala 1.

-Un alt exemplu sunt tabelele ANGAJATI si CONCERTE. Un anagajat poate lucra la mai multe concerte. Pentru a ramane in forma normala 1 am construit tabelul ANGAJATI_CONCERTE.

Un tabel asemanator cu cel de mai jos incalca forma normala 1 deoarece id_concert contine mai multe valori

Id_angajat	Id_concert
10	2,5
4	7,8,9

FN1:

Id_angajat	Id_concert
10	2
10	5
4	7
4	8
4	9

Datorita tabelului ANGAJATI_CONCERTE forma normala 1 este respectata.

FORMA NORMALA 2:

-O relatie se afla in FN2 daca si numai daca aceasta relatie este deja in FN1 , iar fiecare atribut care nu este cheie primara este dependent de intreaga cheie primara.

-Pentru entitatile care au o singura cheie primara (ex: FESTIVALURI, ANGAJATI, CONCERTE) toate atributele sunt automat dependente de cheia primara.

-Tabelele STANDARD,VIP si GOLD (care descriu tipuri de bilete) se afla in forma normala 2. Un exemplu prin care putem observa acest lucru este atributul data_achizitionare. Acest atribut se afla doar in tabela bilete, deoarece depinde doar de id-ul biletului achizitionat, nu si de tipul acestuia. Daca data_achizitionare se afla in interiorul STANDARD, atunci ar fi depins doar partial de cheia primara, deoarece nu depinde si de id_standard.

FORMA NORMALA 3:

- O relatie este in a treia forma normala daca si numai daca este in FN2 si fiecare atribut care nu este cheie depinde direct de cheia primara.

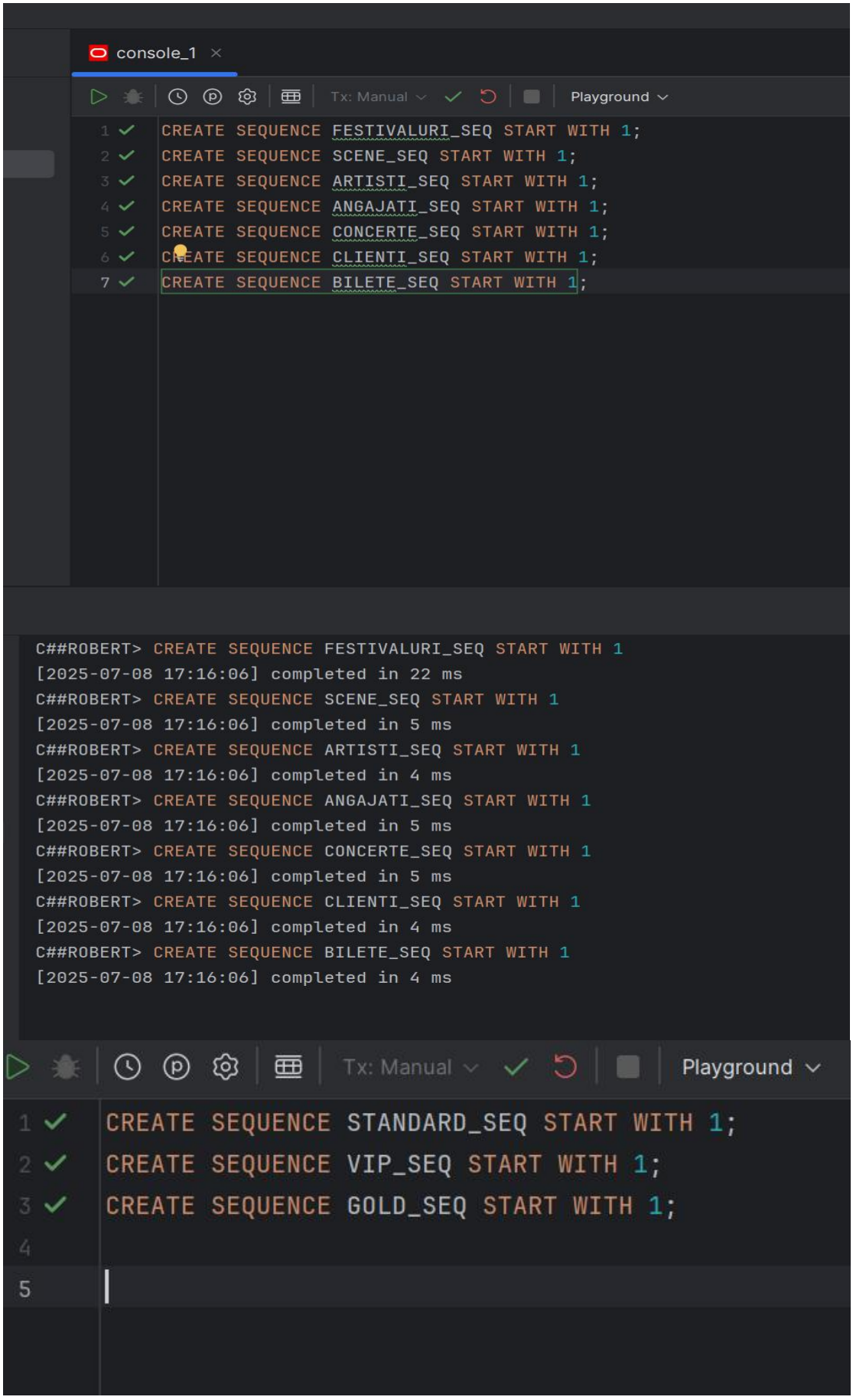
Spre exemplu, in tabela CLIENTI, toate atributele(nume, prenume, email, nr_telefon, data_nastere) depind direct de cheia primara id_client. Acest lucru se intampla in toate tabelele.

10.Crearea unei secvente ce va fi utilizata in inserare inregistrarilor in tabele

Pentru a facilita inserarea valorilor unice în campurile ce reprezinta chei primare, am creat cate o secventa pentru fiecare tabel important din baza de date. Aceste secvente vor genera automat identificatori unici pentru fiecare inregistrare noua.

Cod SQL:

```
CREATE SEQUENCE FESTIVALURI_SEQ START WITH 1;
CREATE SEQUENCE SCENE_SEQ START WITH 1;
CREATE SEQUENCE ARTISTI_SEQ START WITH 1;
CREATE SEQUENCE ANGAJATI_SEQ START WITH 1;
CREATE SEQUENCE CONCERTE_SEQ START WITH 1;
CREATE SEQUENCE CLIENTI_SEQ START WITH 1;
CREATE SEQUENCE STANDAR_SEQ START WITH 1;
CREATE SEQUENCE VIP_SEQ START WITH 1;
CREATE SEQUENCE GOLD_SEQ START WITH 1;
CREATE SEQUENCE BILETE_SEQ START WITH 1;
```


console_1
Tx: Manual
Playground
1 CREATE SEQUENCE FESTIVALURI_SEQ START WITH 1;
2 CREATE SEQUENCE SCENE_SEQ START WITH 1;
3 CREATE SEQUENCE ARTISTI_SEQ START WITH 1;
4 CREATE SEQUENCE ANGAJATI_SEQ START WITH 1;
5 CREATE SEQUENCE CONCERTE_SEQ START WITH 1;
6 CREATE SEQUENCE CLIENTI_SEQ START WITH 1;
7 CREATE SEQUENCE BILETE_SEQ START WITH 1;
C##ROBERT> CREATE SEQUENCE FESTIVALURI_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 22 ms
C##ROBERT> CREATE SEQUENCE SCENE_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 5 ms
C##ROBERT> CREATE SEQUENCE ARTISTI_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 4 ms
C##ROBERT> CREATE SEQUENCE ANGAJATI_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 5 ms
C##ROBERT> CREATE SEQUENCE CONCERTE_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 5 ms
C##ROBERT> CREATE SEQUENCE CLIENTI_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 4 ms
C##ROBERT> CREATE SEQUENCE BILETE_SEQ START WITH 1
[2025-07-08 17:16:06] completed in 4 ms
Tx: Manual
Playground
1 CREATE SEQUENCE STANDARD_SEQ START WITH 1;
2 CREATE SEQUENCE VIP_SEQ START WITH 1;
3 CREATE SEQUENCE GOLD_SEQ START WITH 1;
4
5

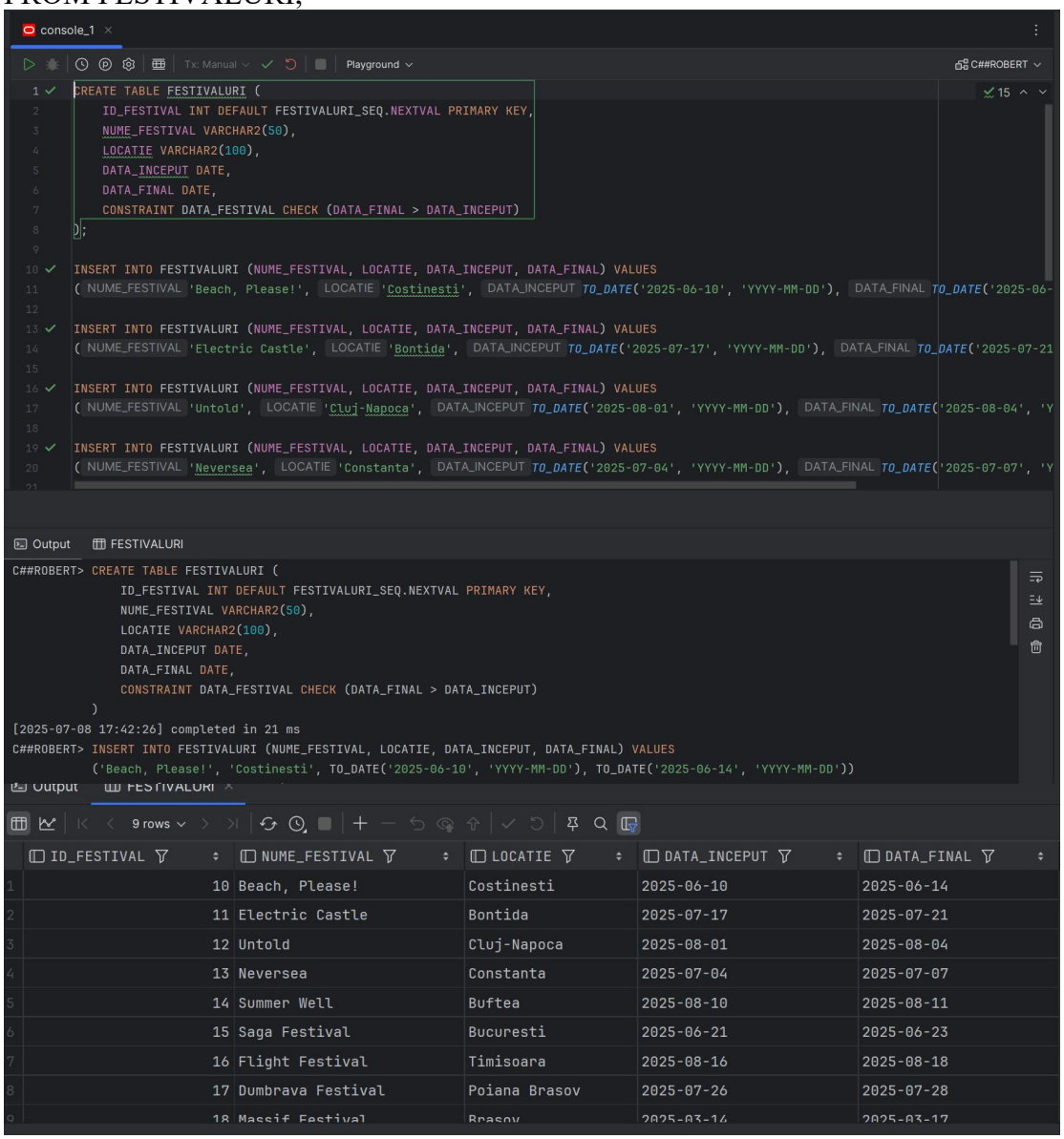
11.Crearea tabelelor SQL si inserarea de date coerente in fiecare dintre acestea

1. Tabelul FESTIVALURI:

Cod SQL:

```
CREATE TABLE FESTIVALURI (
   ID_FESTIVAL INT DEFAULT FESTIVALURI_SEQ.NEXTVAL PRIMARY
KEY,
   NUME_FESTIVAL VARCHAR2(50),
   LOCATIE VARCHAR2(100),
   DATA_INCEPUT DATE,
   DATA_FINAL DATE,
   CONSTRAINT DATA_FESTIVAL CHECK (DATA_FINAL > DATA_INCEPUT)
);

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Beach, Please!', 'Costinesti', TO_DATE('2025-06-10', 'YYYY-MM-DD'),
TO_DATE('2025-06-14', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Electric Castle', 'Bontida', TO_DATE('2025-07-17', 'YYYY-MM-DD'),
TO_DATE('2025-07-21', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Untold', 'Cluj-Napoca', TO_DATE('2025-08-01', 'YYYY-MM-DD'),
TO_DATE('2025-08-04', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Neversea', 'Constanta', TO_DATE('2025-07-04', 'YYYY-MM-DD'),
TO_DATE('2025-07-07', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Summer Well', 'Buftea', TO_DATE('2025-08-10', 'YYYY-MM-DD'),
TO_DATE('2025-08-11', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Saga Festival', 'Bucuresti', TO_DATE('2025-06-21', 'YYYY-MM-DD'),
TO_DATE('2025-06-23', 'YYYY-MM-DD'));
```

```
INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Flight Festival', 'Timisoara', TO_DATE('2025-08-16', 'YYYY-MM-DD'),
TO_DATE('2025-08-18', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Dumbrava Festival', 'Poiana Brasov', TO_DATE('2025-07-26', 'YYYY-MM-DD'),
TO_DATE('2025-07-28', 'YYYY-MM-DD'));

INSERT INTO FESTIVALURI (NUME_FESTIVAL, LOCATIE, DATA_INCEPUT,
DATA_FINAL) VALUES
('Massif Festival', 'Brasov', TO_DATE('2025-03-14', 'YYYY-MM-DD'),
TO_DATE('2025-03-17', 'YYYY-MM-DD'));


SELECT *
FROM FESTIVALURI;
```



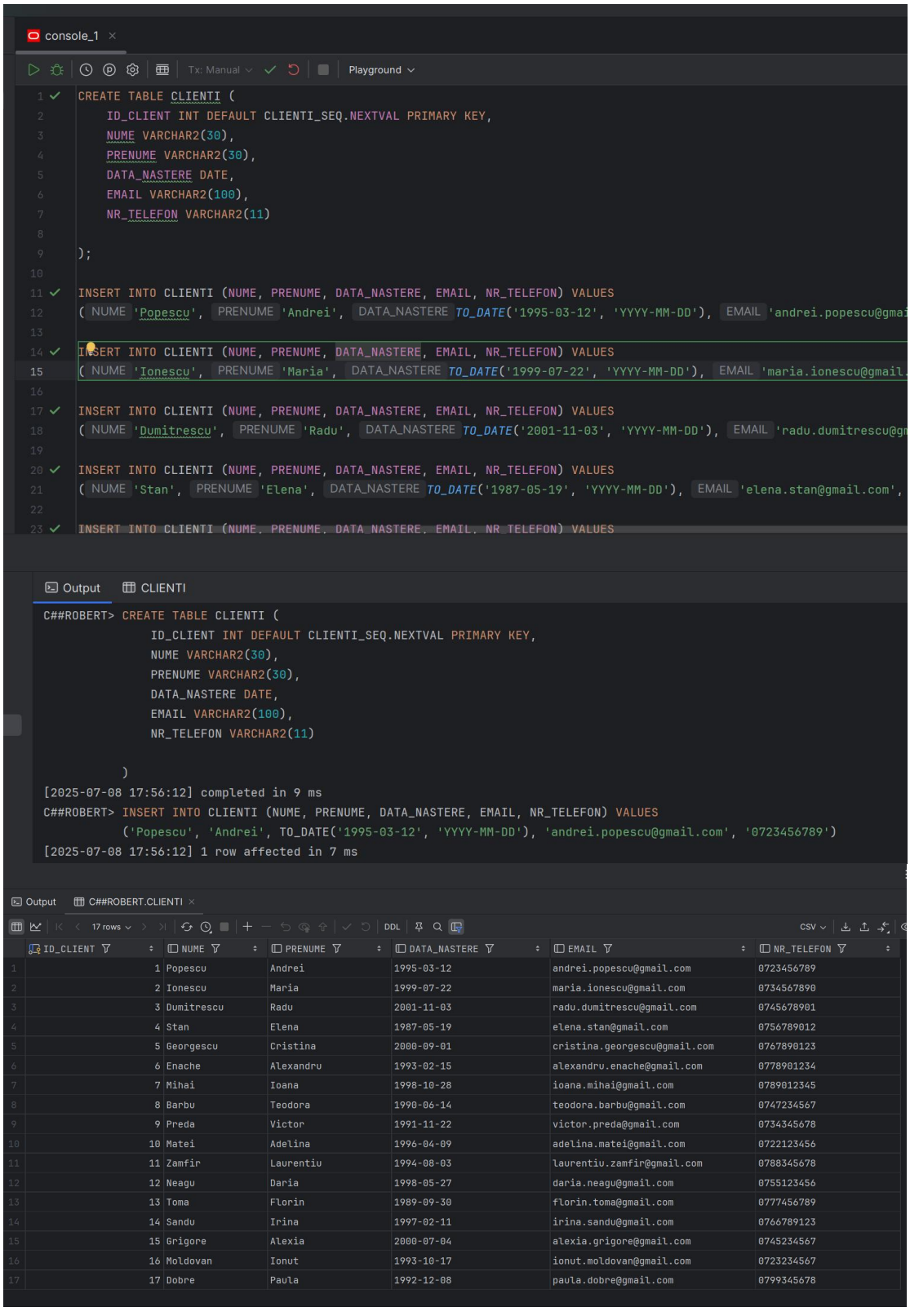
2.Tabelul CLIENTI

Cod SQL:

```
CREATE TABLE CLIENTI (
   ID_CLIENT INT DEFAULT CLIENTI_SEQ.NEXTVAL PRIMARY KEY,
   NUME VARCHAR2(30),
   PRENUME VARCHAR2(30),
   DATA_NASTERE DATE,
   EMAIL VARCHAR2(100),
   NR_TELEFON VARCHAR2(11),

   CONSTRAINT DATA_NASTERE CHECK (DATA_NASTERE < SYSDATE)
);

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Popescu', 'Andrei', TO_DATE('1995-03-12', 'YYYY-MM-DD'),
'andrei.popescu@gmail.com', '0723456789');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Ionescu', 'Maria', TO_DATE('1999-07-22', 'YYYY-MM-DD'),
'maria.ionescu@gmail.com', '0734567890');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Dumitrescu', 'Radu', TO_DATE('2001-11-03', 'YYYY-MM-DD'),
'radu.dumitrescu@gmail.com', '0745678901');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Stan', 'Elena', TO_DATE('1987-05-19', 'YYYY-MM-DD'), 'elena.stan@gmail.com',
'0756789012');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Georgescu', 'Cristina', TO_DATE('2000-09-01', 'YYYY-MM-DD'),
'cristina.georgescu@gmail.com', '0767890123');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Enache', 'Alexandru', TO_DATE('1993-02-15', 'YYYY-MM-DD'),
'alexandru.enache@gmail.com', '0778901234');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON) VALUES
('Mihai', 'Ioana', TO_DATE('1998-10-28', 'YYYY-MM-DD'),
'ioana.mihai@gmail.com', '0789012345');
```

```
INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Barbu', 'Teodora', TO_DATE('1990-06-14', 'YYYY-MM-DD'),
'teodora.barbu@gmail.com', '0747234567');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Preda', 'Victor', TO_DATE('1991-11-22', 'YYYY-MM-DD'),
'victor.preda@gmail.com', '0734345678');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Matei', 'Adelina', TO_DATE('1996-04-09', 'YYYY-MM-DD'),
'adelina.matei@gmail.com', '0722123456');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Zamfir', 'Laurentiu', TO_DATE('1994-08-03', 'YYYY-MM-DD'),
'laurentiu.zamfir@gmail.com', '0788345678');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Neagu', 'Daria', TO_DATE('1998-05-27', 'YYYY-MM-DD'),
'daria.neagu@gmail.com', '0755123456');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Toma', 'Florin', TO_DATE('1989-09-30', 'YYYY-MM-DD'),
'florin.toma@gmail.com', '0777456789');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Sandu', 'Irina', TO_DATE('1997-02-11', 'YYYY-MM-DD'),
'irina.sandu@gmail.com', '0766789123');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Grigore', 'Alexia', TO_DATE('2000-07-04', 'YYYY-MM-DD'),
'alexia.grigore@gmail.com', '0745234567');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Moldovan', 'Ionut', TO_DATE('1993-10-17', 'YYYY-MM-DD'),
'ionut.moldovan@gmail.com', '0723234567');

INSERT INTO CLIENTI (NUME, PRENUME, DATA_NASTERE, EMAIL,
NR_TELEFON)
VALUES ('Dobre', 'Paula', TO_DATE('1992-12-08', 'YYYY-MM-DD'),
'paula.dobre@gmail.com', '0799345678');
```

SELECT *
FROM CLIENTI;



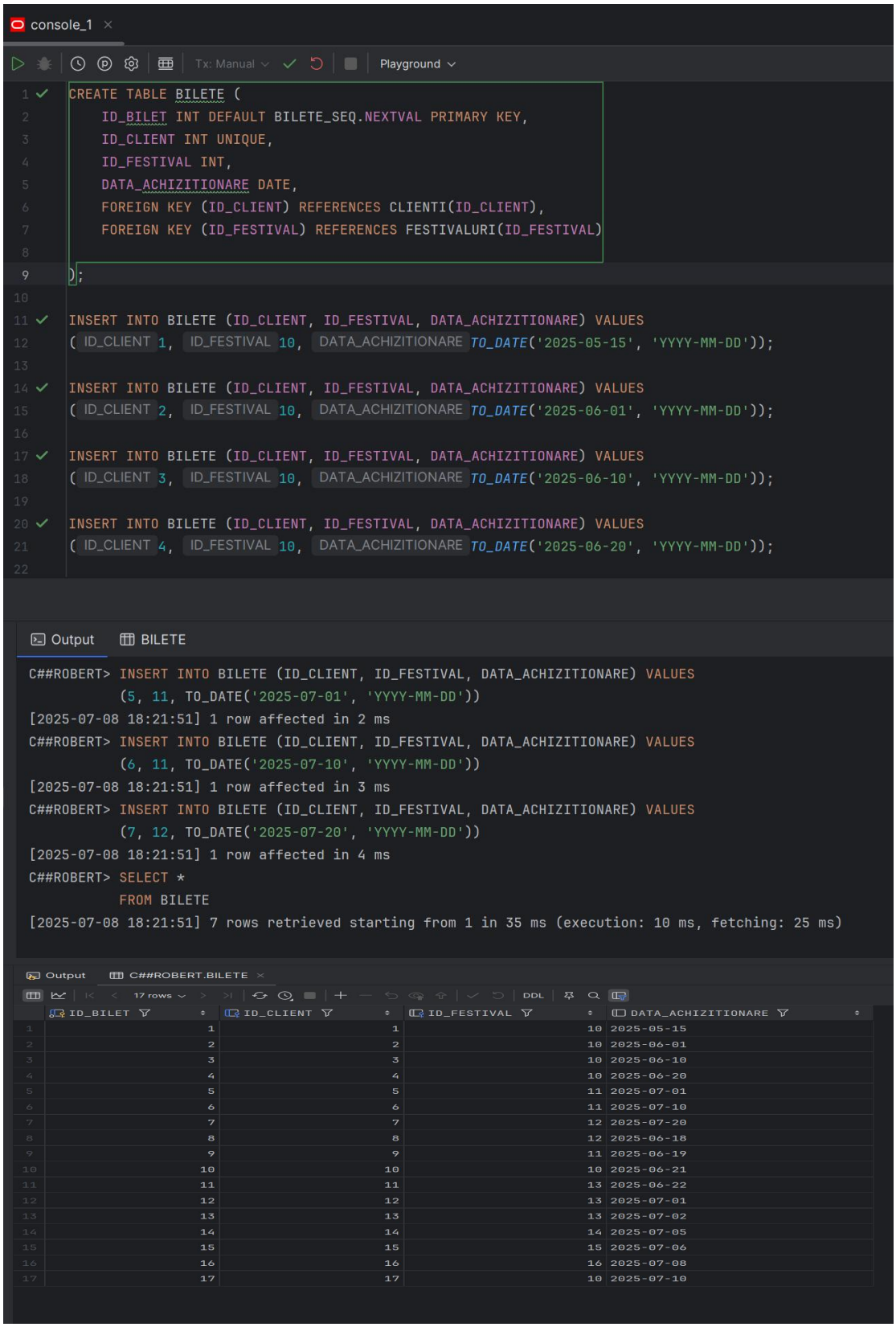
	ID_CLIENT	NUME	PRENUME	DATA_NASTERE	EMAIL	NR_TELEFON
1	1	Popescu	Andrei	1995-03-12	andrei.popescu@gmail.com	0723456789
2	2	Ionescu	Maria	1999-07-22	maria.ionescu@gmail.com	0734567890
3	3	Dumitrescu	Radu	2001-11-03	radu.dumitrescu@gmail.com	0745678901
4	4	Stan	Elena	1987-05-19	elena.stan@gmail.com	0756789012
5	5	Georgescu	Cristina	2000-09-01	cristina.georgescu@gmail.com	0767890123
6	6	Enache	Alexandru	1993-02-15	alexandru.enache@gmail.com	0778901234
7	7	Mihai	Ioana	1998-10-28	ioana.mihai@gmail.com	0789012345
8	8	Barbu	Teodora	1990-06-14	teodora.barbu@gmail.com	0747234567
9	9	Preda	Victor	1991-11-22	victor.preda@gmail.com	0734345678
10	10	Matei	Adelina	1996-04-09	adelina.matei@gmail.com	0722123456
11	11	Zamfir	Laurentiu	1994-08-03	laurentiu.zamfir@gmail.com	0788345678
12	12	Neagu	Daria	1998-05-27	daria.neagu@gmail.com	0755123456
13	13	Toma	Florin	1989-09-30	florin.toma@gmail.com	0777456789
14	14	Sandu	Irina	1997-02-11	irina.sandu@gmail.com	0766789123
15	15	Grigore	Alexia	2000-07-04	alexia.grigore@gmail.com	0745234567
16	16	Moldovan	Ionut	1993-10-17	ionut.moldovan@gmail.com	0723234567
17	17	Dobre	Paula	1992-12-08	paula.dobre@gmail.com	0799345678

3.Tabela BILETE

Cod SQL:

```
CREATE TABLE BILETE (
   ID_BILET INT DEFAULT BILETE_SEQ.NEXTVAL PRIMARY KEY,
   ID_CLIENT INT UNIQUE,
   ID_FESTIVAL INT,
   DATA_ACHIZITIONARE DATE,
   FOREIGN KEY (ID_CLIENT) REFERENCES CLIENTI(ID_CLIENT),
   FOREIGN KEY (ID_FESTIVAL) REFERENCES
FESTIVALURI(ID_FESTIVAL)

);

INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (1, 10, TO_DATE('2025-05-15', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (2, 10, TO_DATE('2025-06-01', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (3, 10, TO_DATE('2025-06-10', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (4, 10, TO_DATE('2025-06-20', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (5, 11, TO_DATE('2025-07-01', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (6, 11, TO_DATE('2025-07-10', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (7, 12, TO_DATE('2025-07-20', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (8, 12, TO_DATE('2025-06-18', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (9, 11, TO_DATE('2025-06-19', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (10, 10, TO_DATE('2025-06-21', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (11, 13, TO_DATE('2025-06-22', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (12, 13, TO_DATE('2025-07-01', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (13, 13, TO_DATE('2025-07-02', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (14, 14, TO_DATE('2025-07-05', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (15, 15, TO_DATE('2025-07-06', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (16, 16, TO_DATE('2025-07-08', 'YYYY-MM-DD'));
INSERT INTO BILETE (ID_CLIENT, ID_FESTIVAL, DATA_ACHIZITIONARE)
VALUES (17, 10, TO_DATE('2025-07-10', 'YYYY-MM-DD'));
```

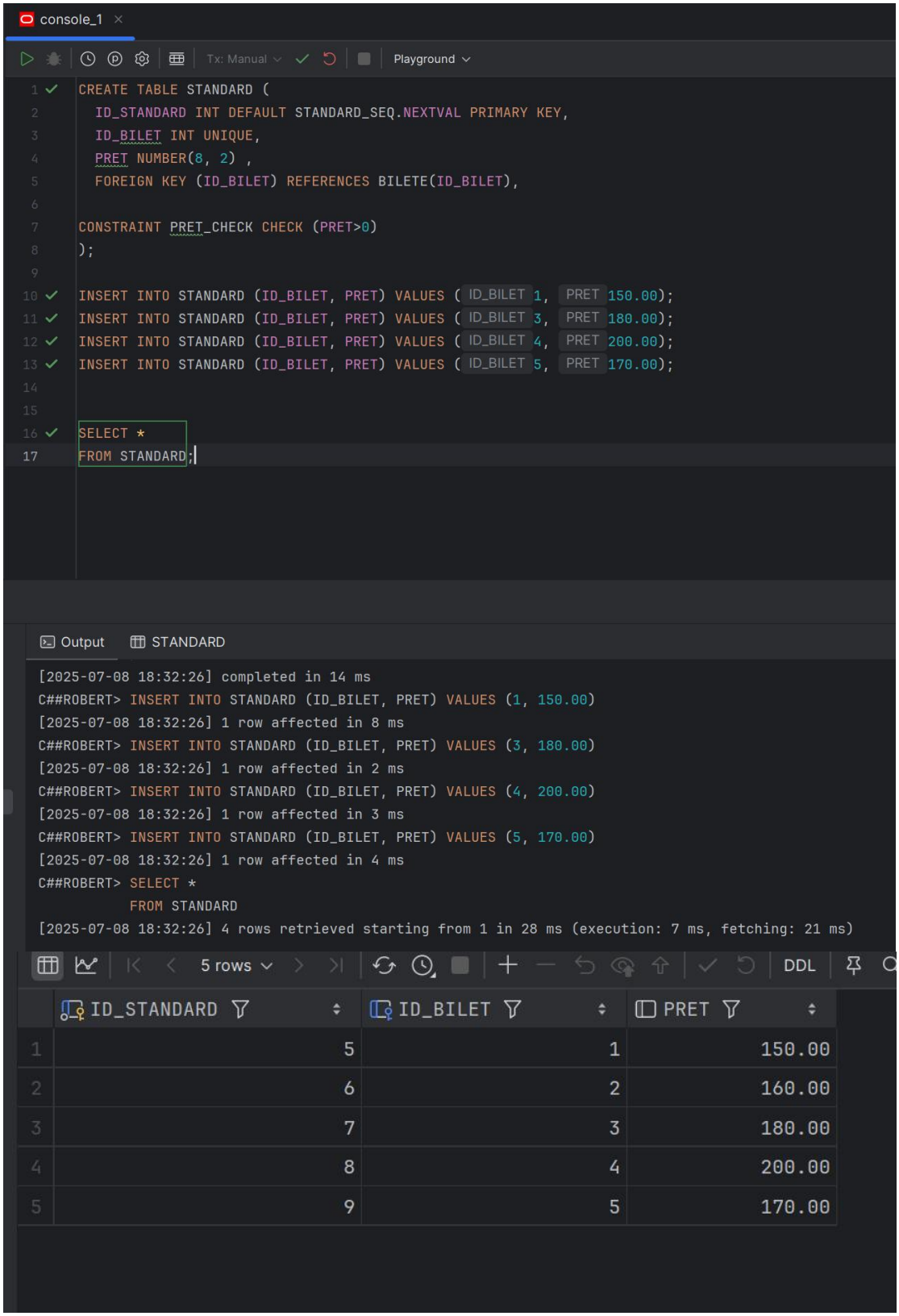
SELECT *
FROM BILETE;



4.Tabela STANDARD

Cod SQL:

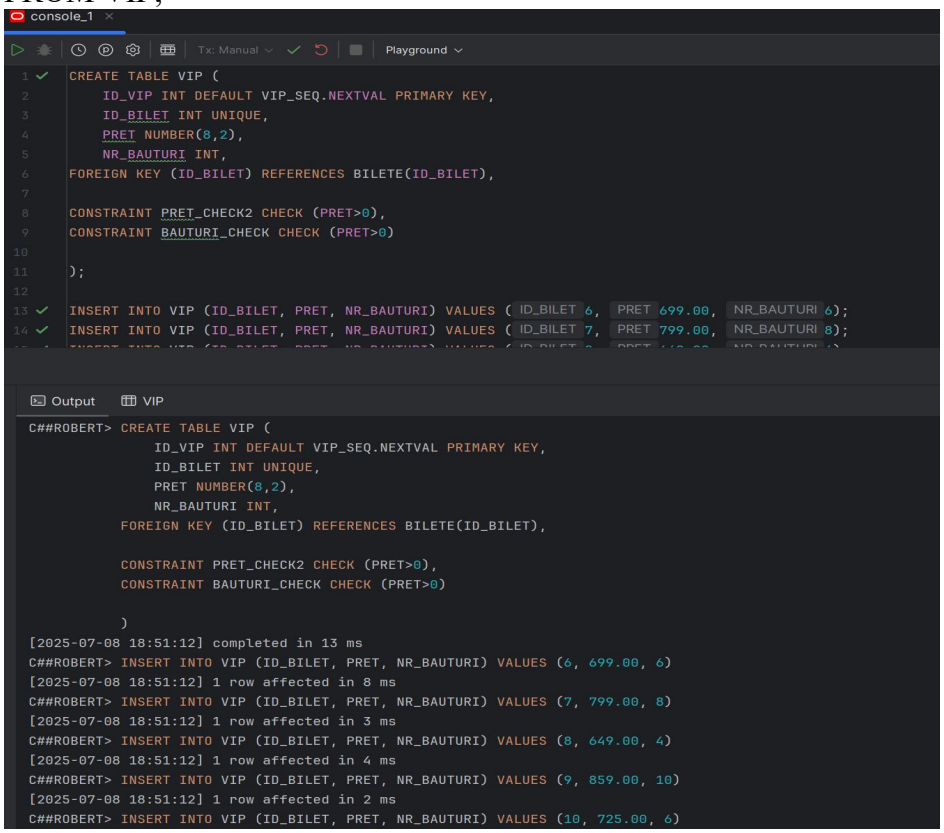
```
CREATE TABLE STANDARD (
  ID_STANDARD INT DEFAULT STANDARD_SEQ.NEXTVAL PRIMARY KEY,
  ID_BILET INT UNIQUE,
  PRET NUMBER(8, 2) ,
  FOREIGN KEY (ID_BILET) REFERENCES BILETE(ID_BILET),

CONSTRAINT PRET_CHECK CHECK (PRET>0)
);

INSERT INTO STANDARD (ID_BILET, PRET) VALUES (1, 150.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (2, 160.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (3, 180.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (4, 200.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (5, 170.00);


SELECT *
FROM STANDARD
```

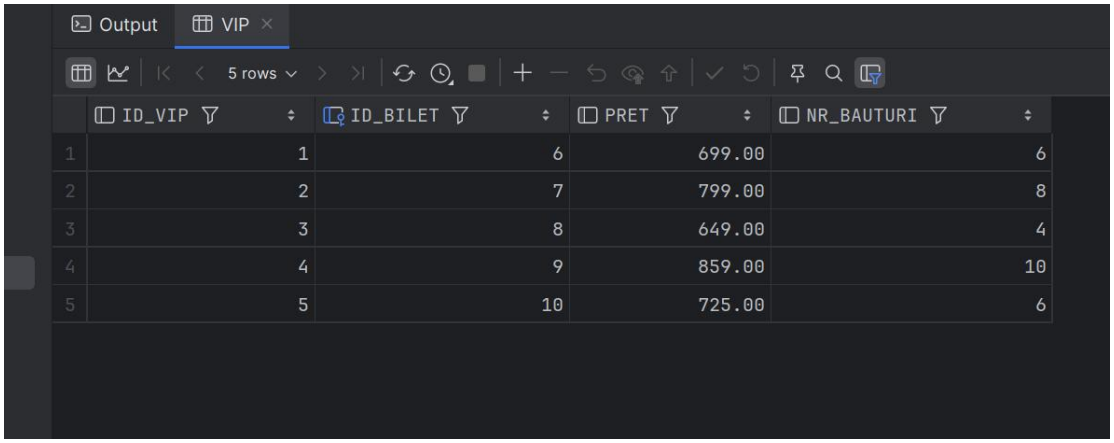

console_1
Tx: Manual
Playground
CREATE TABLE STANDARD (
ID_STANDARD INT DEFAULT STANDARD_SEQ.NEXTVAL PRIMARY KEY,
ID_BILET INT UNIQUE,
PRET NUMBER(8, 2) ,
FOREIGN KEY (ID_BILET) REFERENCES BILETE(ID_BILET),
CONSTRAINT PRET_CHECK CHECK (PRET>0)
);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (ID_BILET 1, PRET 150.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (ID_BILET 3, PRET 180.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (ID_BILET 4, PRET 200.00);
INSERT INTO STANDARD (ID_BILET, PRET) VALUES (ID_BILET 5, PRET 170.00);
SELECT *
FROM STANDARD;
Output
STANDARD
[2025-07-08 18:32:26] completed in 14 ms
C##ROBERT> INSERT INTO STANDARD (ID_BILET, PRET) VALUES (1, 150.00)
[2025-07-08 18:32:26] 1 row affected in 8 ms
C##ROBERT> INSERT INTO STANDARD (ID_BILET, PRET) VALUES (3, 180.00)
[2025-07-08 18:32:26] 1 row affected in 2 ms
C##ROBERT> INSERT INTO STANDARD (ID_BILET, PRET) VALUES (4, 200.00)
[2025-07-08 18:32:26] 1 row affected in 3 ms
C##ROBERT> INSERT INTO STANDARD (ID_BILET, PRET) VALUES (5, 170.00)
[2025-07-08 18:32:26] 1 row affected in 4 ms
C##ROBERT> SELECT *
FROM STANDARD
[2025-07-08 18:32:26] 4 rows retrieved starting from 1 in 28 ms (execution: 7 ms, fetching: 21 ms)
5 rows
DDL
ID_STANDARD
ID_BILET
PRET
1 5 1 150.00
2 6 2 160.00
3 7 3 180.00
4 8 4 200.00
5 9 5 170.00

5.Tabela VIP

Cod sql:

```
CREATE TABLE VIP (
   ID_VIP INT DEFAULT VIP_SEQ.NEXTVAL PRIMARY KEY,
   ID_BILET INT UNIQUE,
   PRET NUMBER(8,2),
   NR_BAUTURI INT,
  FOREIGN KEY (ID_BILET) REFERENCES BILETE(ID_BILET),

CONSTRAINT PRET_CHECK2 CHECK (PRET>0),
CONSTRAINT BAUTURI_CHECK CHECK (PRET>0)

);

INSERT INTO VIP (ID_BILET, PRET, NR_BAUTURI) VALUES (6, 699.00, 6);
INSERT INTO VIP (ID_BILET, PRET, NR_BAUTURI) VALUES (7, 799.00, 8);
INSERT INTO VIP (ID_BILET, PRET, NR_BAUTURI) VALUES (8, 649.00, 4);
INSERT INTO VIP (ID_BILET, PRET, NR_BAUTURI) VALUES (9, 859.00, 10);
INSERT INTO VIP (ID_BILET, PRET, NR_BAUTURI) VALUES (10, 725.00, 6);


SELECT *
FROM VIP;
```



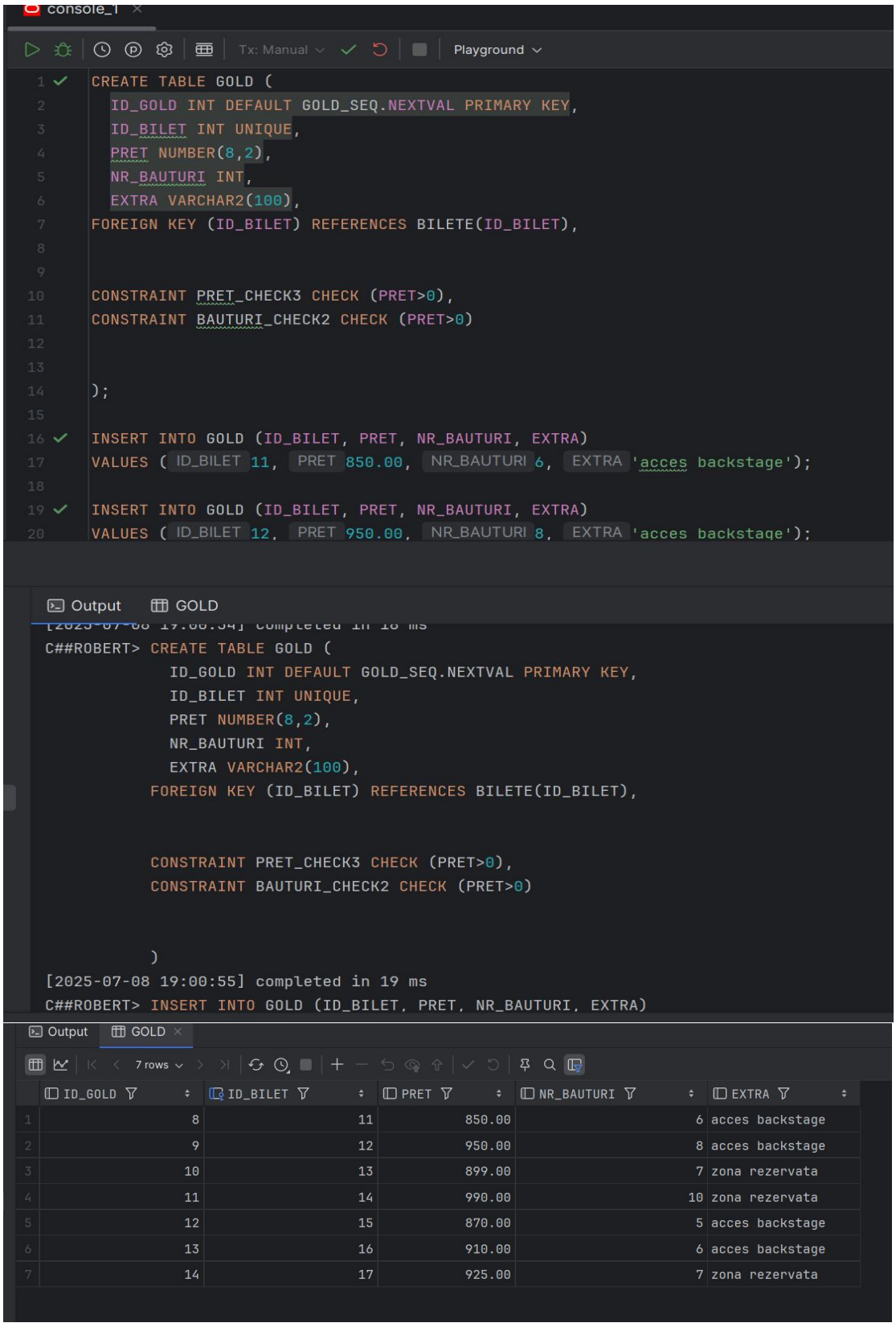


6. Tabela GOLD

```
CREATE TABLE GOLD (
  ID_GOLD INT DEFAULT GOLD_SEQ.NEXTVAL PRIMARY KEY,
  ID_BILET INT UNIQUE,
  PRET NUMBER(8,2),
  NR_BAUTURI INT,
  EXTRA VARCHAR2(100),
FOREIGN KEY (ID_BILET) REFERENCES BILETE(ID_BILET),


CONSTRAINT PRET_CHECK3 CHECK (PRET>0),
CONSTRAINT BAUTURI_CHECK2 CHECK (PRET>0)


);

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
VALUES (11, 850.00, 6, 'acces backstage');

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
VALUES (12, 950.00, 8, 'acces backstage');

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
VALUES (13, 899.00, 7, 'zona rezervata');

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
VALUES (14, 990.00, 10, 'zona rezervata');

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
VALUES (15, 870.00, 5,  'acces backstage');

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
VALUES (16, 910.00, 6, 'acces backstage');

INSERT INTO GOLD (ID_BILET, PRET, NR_BAUTURI, EXTRA)
```

VALUES (17, 925.00, 7,'zona rezervata');

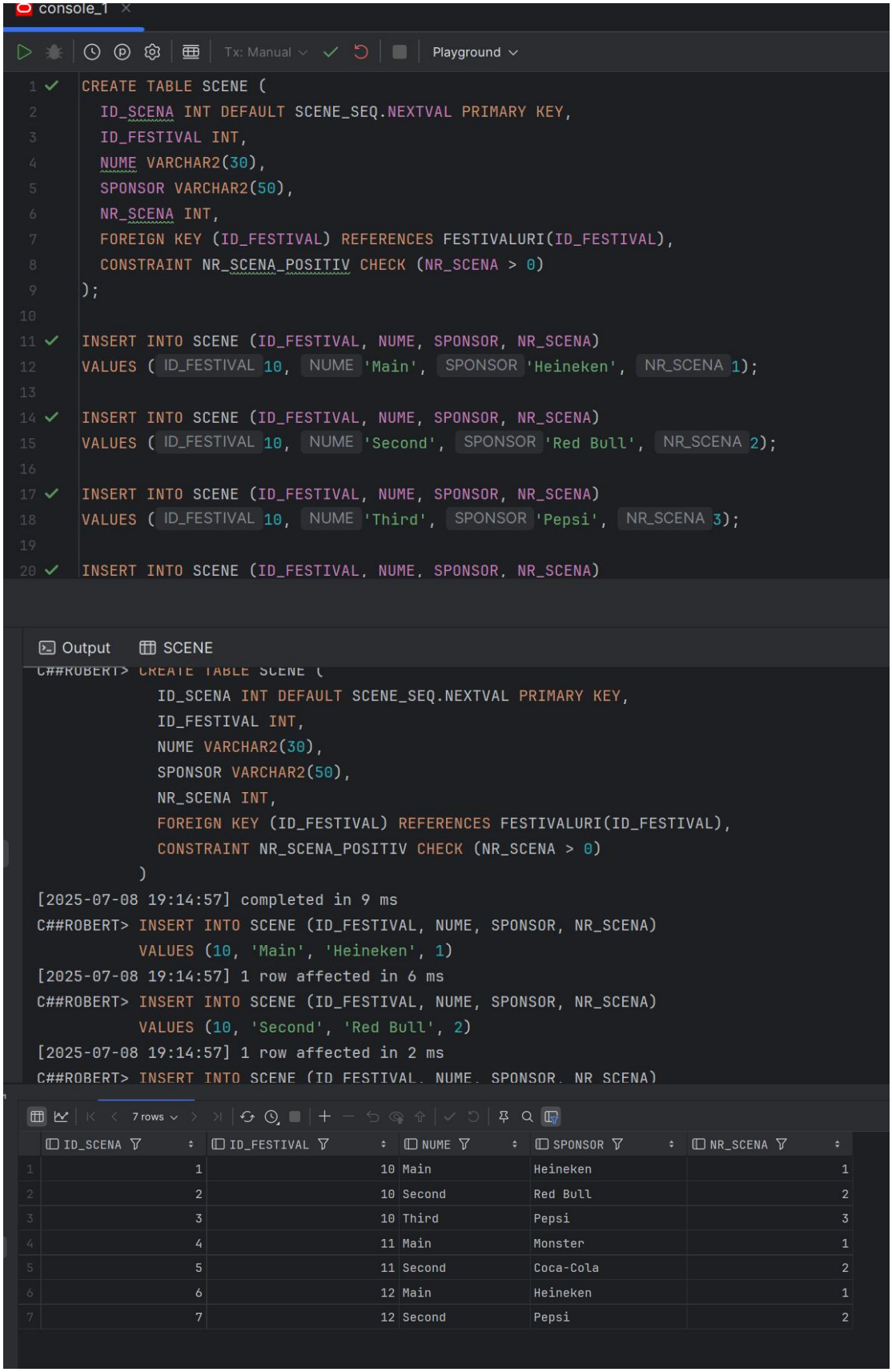
SELECT *
FROM GOLD;



7.Tabela SCENE

Cod SQL:

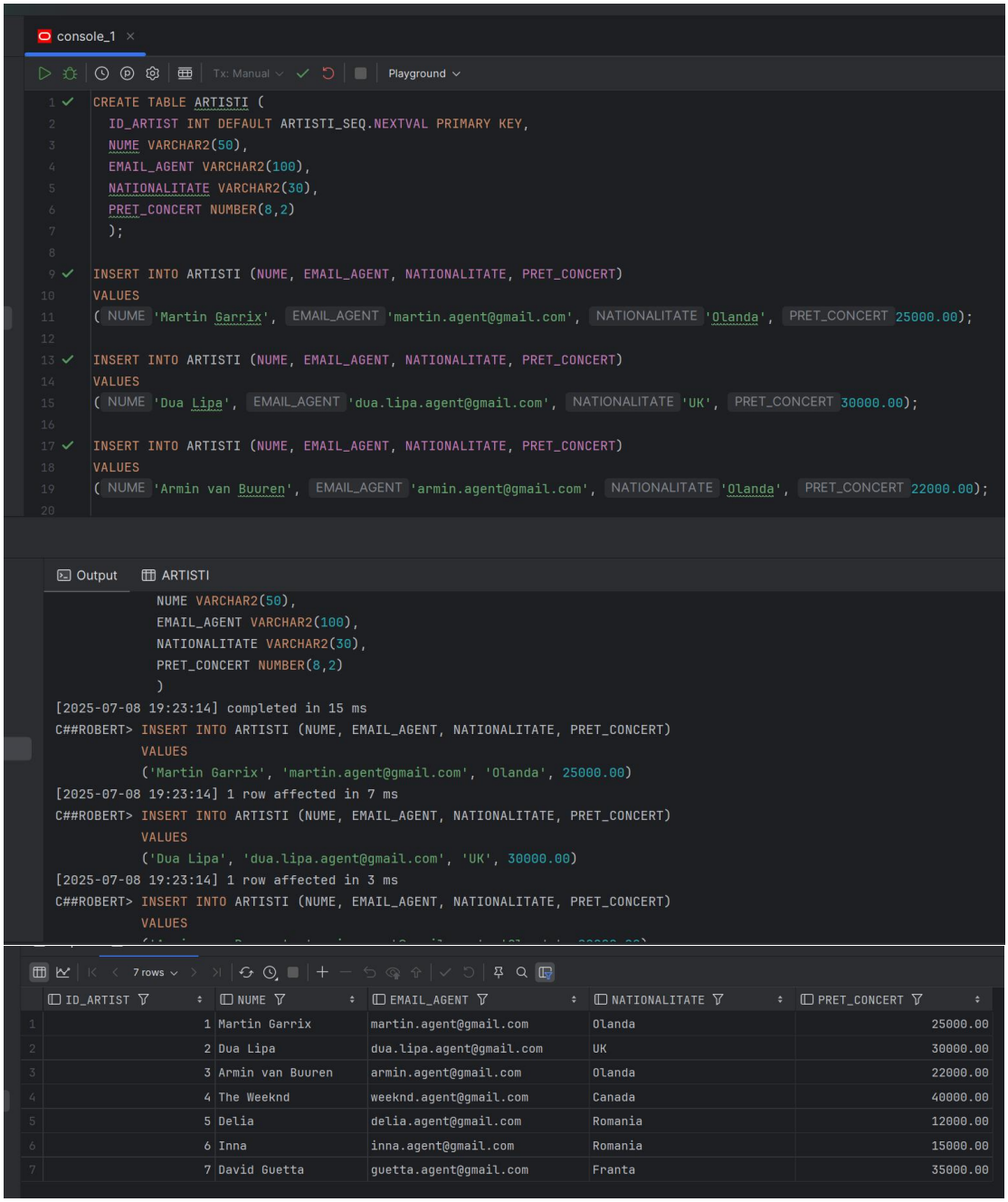
```
CREATE TABLE SCENE (
  ID_SCENA INT DEFAULT SCENE_SEQ.NEXTVAL PRIMARY KEY,
  ID_FESTIVAL INT,
  NUME VARCHAR2(30),
  SPONSOR VARCHAR2(50),
  NR_SCENA INT,
  FOREIGN KEY (ID_FESTIVAL) REFERENCES FESTIVALURI(ID_FESTIVAL),
  CONSTRAINT NR_SCENA_POSITIV CHECK (NR_SCENA > 0)
);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (10, 'Main', 'Heineken', 1);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (10, 'Second', 'Red Bull', 2);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (10, 'Third', 'Pepsi', 3);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (11, 'Main', 'Monster', 1);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (11, 'Second', 'Coca-Cola', 2);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (12, 'Main', 'Heineken', 1);

INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (12, 'Second', 'Pepsi', 2);

SELECT *
FROM SCENE;
```


console_1
Tx: Manual
Playground
CREATE TABLE SCENE (
ID_SCENA INT DEFAULT SCENE_SEQ.NEXTVAL PRIMARY KEY,
ID_FESTIVAL INT,
NUME VARCHAR2(30),
SPONSOR VARCHAR2(50),
NR_SCENA INT,
FOREIGN KEY (ID_FESTIVAL) REFERENCES FESTIVALURI(ID_FESTIVAL),
CONSTRAINT NR_SCENA_POSITIV CHECK (NR_SCENA > 0)
);
INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (ID_FESTIVAL 10, NUME 'Main', SPONSOR 'Heineken', NR_SCENA 1);
INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (ID_FESTIVAL 10, NUME 'Second', SPONSOR 'Red Bull', NR_SCENA 2);
INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (ID_FESTIVAL 10, NUME 'Third', SPONSOR 'Pepsi', NR_SCENA 3);
INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
Output
SCENE
C##ROBERT> CREATE TABLE SCENE (
ID_SCENA INT DEFAULT SCENE_SEQ.NEXTVAL PRIMARY KEY,
ID_FESTIVAL INT,
NUME VARCHAR2(30),
SPONSOR VARCHAR2(50),
NR_SCENA INT,
FOREIGN KEY (ID_FESTIVAL) REFERENCES FESTIVALURI(ID_FESTIVAL),
CONSTRAINT NR_SCENA_POSITIV CHECK (NR_SCENA > 0)
)
[2025-07-08 19:14:57] completed in 9 ms
C##ROBERT> INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (10, 'Main', 'Heineken', 1)
[2025-07-08 19:14:57] 1 row affected in 6 ms
C##ROBERT> INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
VALUES (10, 'Second', 'Red Bull', 2)
[2025-07-08 19:14:57] 1 row affected in 2 ms
C##ROBERT> INSERT INTO SCENE (ID_FESTIVAL, NUME, SPONSOR, NR_SCENA)
7 rows
ID_SCENA ID_FESTIVAL NUME SPONSOR NR_SCENA
1 1 10 Main Heineken 1
2 2 10 Second Red Bull 2
3 3 10 Third Pepsi 3
4 4 11 Main Monster 1
5 5 11 Second Coca-Cola 2
6 6 12 Main Heineken 1
7 7 12 Second Pepsi 2

8.Tabela ARTISTI

Cod SQL:

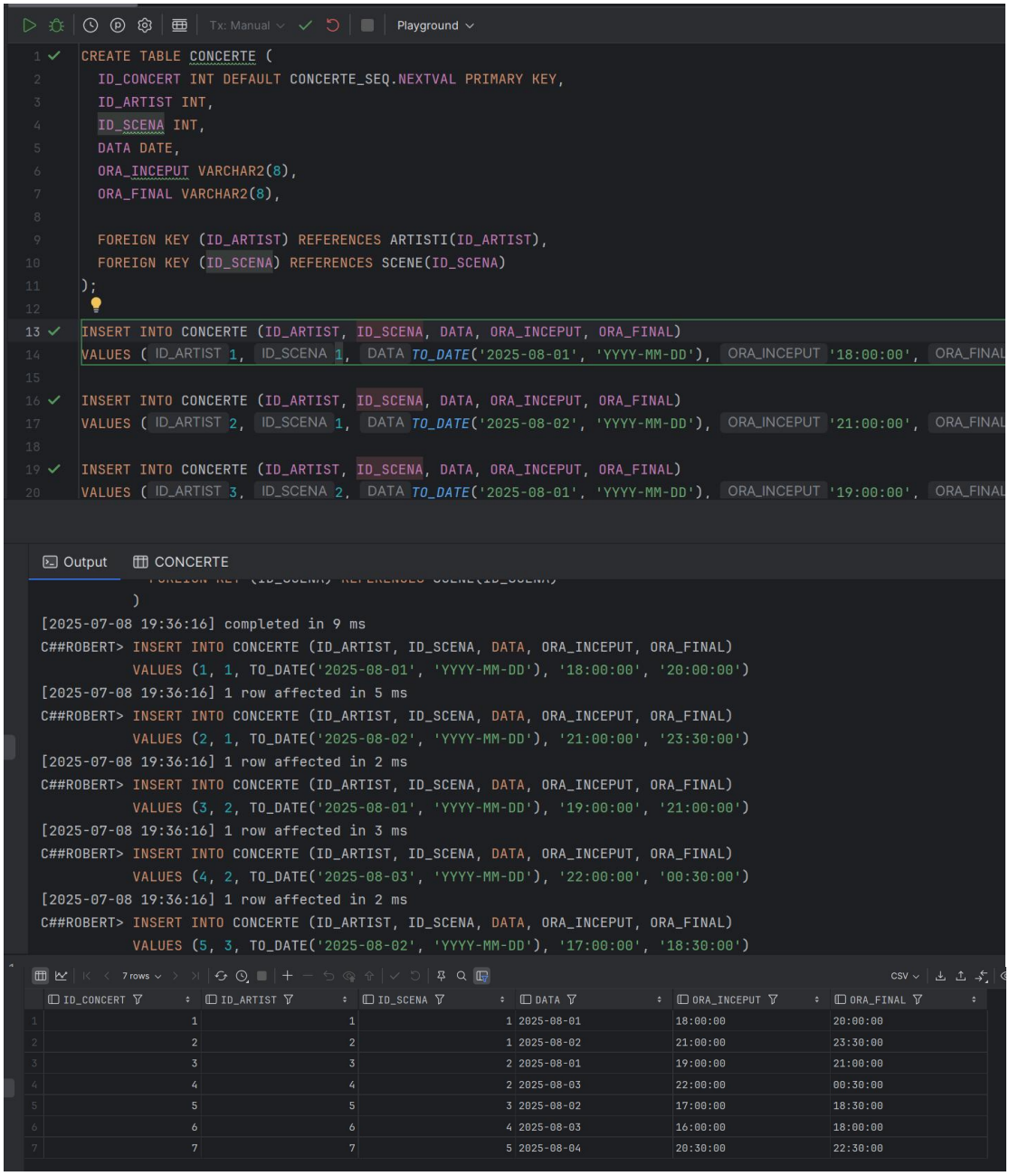
```
CREATE TABLE ARTISTI (
  ID_ARTIST INT DEFAULT ARTISTI_SEQ.NEXTVAL PRIMARY KEY,
  NUME VARCHAR2(50),
  EMAIL_AGENT VARCHAR2(100),
  NATIONALITATE VARCHAR2(30),
  PRET_CONCERT NUMBER(8,2)
  );

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
('Martin Garrix', 'martin.agent@gmail.com', 'Olanda', 25000.00);

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
('Dua Lipa', 'dua.lipa.agent@gmail.com', 'UK', 30000.00);

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
('Armin van Buuren', 'armin.agent@gmail.com', 'Olanda', 22000.00);

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
 ('The Weeknd', 'weeknd.agent@gmail.com', 'Canada', 40000.00);

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
('Delia', 'delia.agent@gmail.com', 'Romania', 12000.00);

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
('Inna', 'inna.agent@gmail.com', 'Romania', 15000.00);

INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE,
PRET_CONCERT)
VALUES
('David Guetta', 'guetta.agent@gmail.com', 'Franta', 35000.00);

SELECT *
FROM ARTISTI;
```


console_1
Tx: Manual
Playground
CREATE TABLE ARTISTI (
ID_ARTIST INT DEFAULT ARTISTI_SEQ.NEXTVAL PRIMARY KEY,
NUME VARCHAR2(50),
EMAIL_AGENT VARCHAR2(100),
NATIONALITATE VARCHAR2(30),
PRET_CONCERT NUMBER(8,2)
);
INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE, PRET_CONCERT)
VALUES
(NUME 'Martin Garrix', EMAIL_AGENT 'martin.agent@gmail.com', NATIONALITATE 'Olanda', PRET_CONCERT 25000.00);
INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE, PRET_CONCERT)
VALUES
(NUME 'Dua Lipa', EMAIL_AGENT 'dua.lipa.agent@gmail.com', NATIONALITATE 'UK', PRET_CONCERT 30000.00);
INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE, PRET_CONCERT)
VALUES
(NUME 'Armin van Buuren', EMAIL_AGENT 'armin.agent@gmail.com', NATIONALITATE 'Olanda', PRET_CONCERT 22000.00);
Output
ARTISTI
NUME VARCHAR2(50),
EMAIL_AGENT VARCHAR2(100),
NATIONALITATE VARCHAR2(30),
PRET_CONCERT NUMBER(8,2)
)
[2025-07-08 19:23:14] completed in 15 ms
C##ROBERT> INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE, PRET_CONCERT)
VALUES
('Martin Garrix', 'martin.agent@gmail.com', 'Olanda', 25000.00)
[2025-07-08 19:23:14] 1 row affected in 7 ms
C##ROBERT> INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE, PRET_CONCERT)
VALUES
('Dua Lipa', 'dua.lipa.agent@gmail.com', 'UK', 30000.00)
[2025-07-08 19:23:14] 1 row affected in 3 ms
C##ROBERT> INSERT INTO ARTISTI (NUME, EMAIL_AGENT, NATIONALITATE, PRET_CONCERT)
VALUES
7 rows
ID_ARTIST NUME EMAIL_AGENT NATIONALITATE PRET_CONCERT
1 1 Martin Garrix martin.agent@gmail.com Olanda 25000.00
2 2 Dua Lipa dua.lipa.agent@gmail.com UK 30000.00
3 3 Armin van Buuren armin.agent@gmail.com Olanda 22000.00
4 4 The Weeknd weeknd.agent@gmail.com Canada 40000.00
5 5 Delia delia.agent@gmail.com Romania 12000.00
6 6 Inna inna.agent@gmail.com Romania 15000.00
7 7 David Guetta guetta.agent@gmail.com Franta 35000.00

9.Tabela CONCERTE

Cod SQL:

```
CREATE TABLE CONCERTE (
  ID_CONCERT INT DEFAULT CONCERTE_SEQ.NEXTVAL PRIMARY KEY,
  ID_ARTIST INT,
  ID_SCENA INT,
  DATA DATE,
  ORA_INCEPUT VARCHAR2(8),
  ORA_FINAL VARCHAR2(8),

  FOREIGN KEY (ID_ARTIST) REFERENCES ARTISTI(ID_ARTIST),
  FOREIGN KEY (ID_SCENA) REFERENCES SCENE(ID_SCENA)
);

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (1, 1, TO_DATE('2025-08-01', 'YYYY-MM-DD'), '18:00:00', '20:00:00');

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (2, 1, TO_DATE('2025-08-02', 'YYYY-MM-DD'), '21:00:00', '23:30:00');

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (3, 2, TO_DATE('2025-08-01', 'YYYY-MM-DD'), '19:00:00', '21:00:00');

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (4, 2, TO_DATE('2025-08-03', 'YYYY-MM-DD'), '22:00:00', '00:30:00');

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (5, 3, TO_DATE('2025-08-02', 'YYYY-MM-DD'), '17:00:00', '18:30:00');

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (6, 4, TO_DATE('2025-08-03', 'YYYY-MM-DD'), '16:00:00', '18:00:00');

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT,
ORA_FINAL)
VALUES (7, 5, TO_DATE('2025-08-04', 'YYYY-MM-DD'), '20:30:00', '22:30:00');


SELECT *
FROM CONCERTE;
```



```
CREATE TABLE CONCERTE (
  ID_CONCERT INT DEFAULT CONCERTE_SEQ.NEXTVAL PRIMARY KEY,
  ID_ARTIST INT,
  ID_SCENA INT,
  DATA DATE,
  ORA_INCEPUT VARCHAR2(8),
  ORA_FINAL VARCHAR2(8),

  FOREIGN KEY (ID_ARTIST) REFERENCES ARTISTI(ID_ARTIST),
  FOREIGN KEY (ID_SCENA) REFERENCES SCENE(ID_SCENA)
);

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
VALUES ( ID_ARTIST 1, ID_SCENA 1, DATA TO_DATE('2025-08-01', 'YYYY-MM-DD'), ORA_INCEPUT '18:00:00', ORA_FINAL

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
VALUES ( ID_ARTIST 2, ID_SCENA 1, DATA TO_DATE('2025-08-02', 'YYYY-MM-DD'), ORA_INCEPUT '21:00:00', ORA_FINAL

INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
VALUES ( ID_ARTIST 3, ID_SCENA 2, DATA TO_DATE('2025-08-01', 'YYYY-MM-DD'), ORA_INCEPUT '19:00:00', ORA_FINAL
```

Output CONCERTE

```
          )
[2025-07-08 19:36:16] completed in 9 ms
C##ROBERT> INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
           VALUES (1, 1, TO_DATE('2025-08-01', 'YYYY-MM-DD'), '18:00:00', '20:00:00')
[2025-07-08 19:36:16] 1 row affected in 5 ms
C##ROBERT> INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
           VALUES (2, 1, TO_DATE('2025-08-02', 'YYYY-MM-DD'), '21:00:00', '23:30:00')
[2025-07-08 19:36:16] 1 row affected in 2 ms
C##ROBERT> INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
           VALUES (3, 2, TO_DATE('2025-08-01', 'YYYY-MM-DD'), '19:00:00', '21:00:00')
[2025-07-08 19:36:16] 1 row affected in 3 ms
C##ROBERT> INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
           VALUES (4, 2, TO_DATE('2025-08-03', 'YYYY-MM-DD'), '22:00:00', '00:30:00')
[2025-07-08 19:36:16] 1 row affected in 2 ms
C##ROBERT> INSERT INTO CONCERTE (ID_ARTIST, ID_SCENA, DATA, ORA_INCEPUT, ORA_FINAL)
           VALUES (5, 3, TO_DATE('2025-08-02', 'YYYY-MM-DD'), '17:00:00', '18:30:00')
```

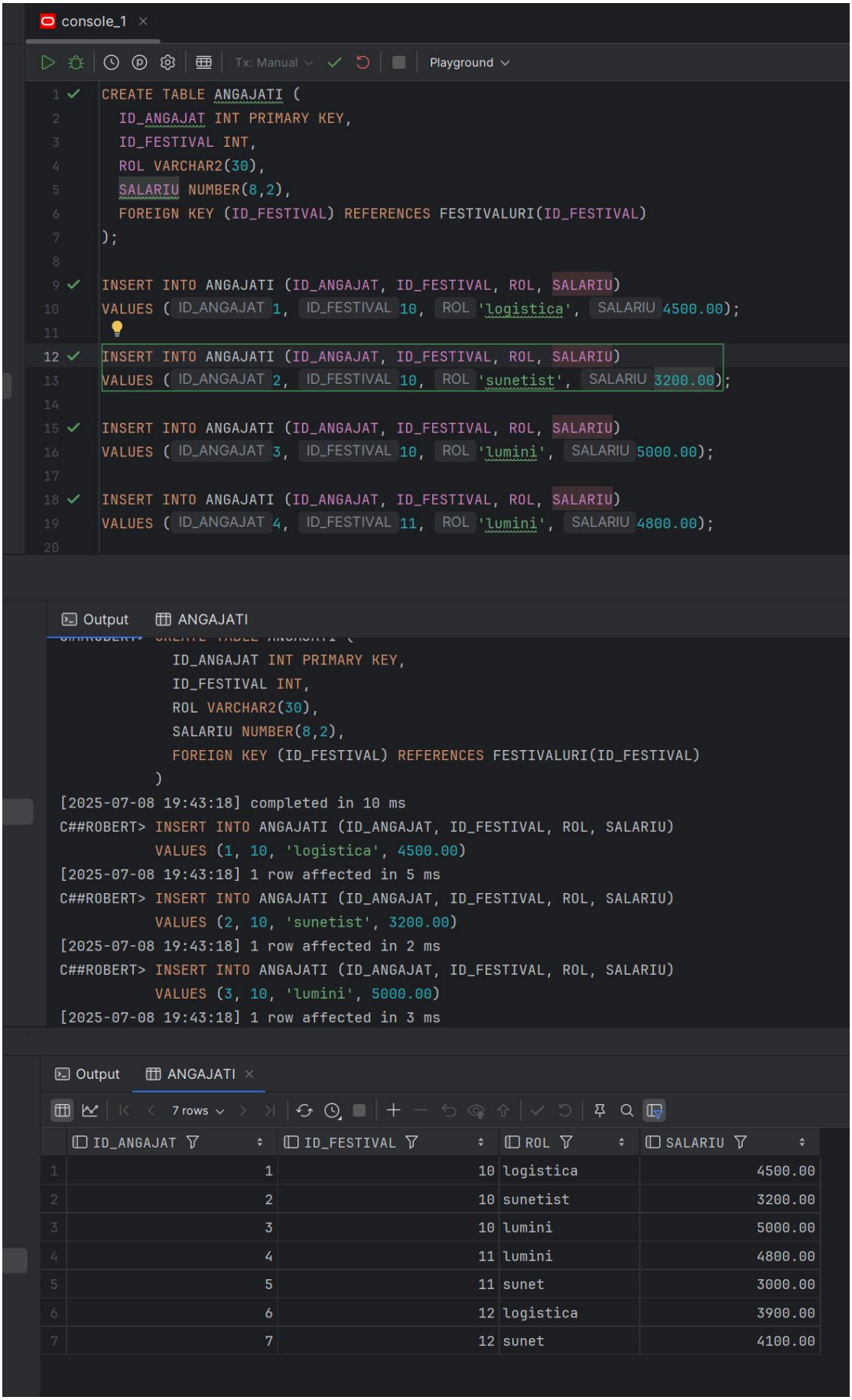
7 rows CSV

	ID_CONCERT	ID_ARTIST	ID_SCENA	DATA	ORA_INCEPUT	ORA_FINAL
1	1	1	1	2025-08-01	18:00:00	20:00:00
2	2	2	1	2025-08-02	21:00:00	23:30:00
3	3	3	2	2025-08-01	19:00:00	21:00:00
4	4	4	2	2025-08-03	22:00:00	00:30:00
5	5	5	3	2025-08-02	17:00:00	18:30:00
6	6	6	4	2025-08-03	16:00:00	18:00:00
7	7	7	5	2025-08-04	20:30:00	22:30:00

10.Tabela angajati

Cod SQL:

```
CREATE TABLE ANGAJATI (
  ID_ANGAJAT INT PRIMARY KEY,
  ID_FESTIVAL INT,
  ROL VARCHAR2(30),
  SALARIU NUMBER(8,2),
  FOREIGN KEY (ID_FESTIVAL) REFERENCES FESTIVALURI(ID_FESTIVAL)
);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (1, 10, 'logistica', 4500.00);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (2, 10, 'sunetist', 3200.00);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (3, 10, 'lumini', 5000.00);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (4, 11, 'lumini', 4800.00);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (5, 11, 'sunet',3000.00);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (6, 12, 'logistica', 3900.00);

INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (7, 12, 'sunet', 4100.00);

SELECT *
FROM ANGAJATI;
```


console_1
Tx: Manual
Playground
CREATE TABLE ANGAJATI (
ID_ANGAJAT INT PRIMARY KEY,
ID_FESTIVAL INT,
ROL VARCHAR2(30),
SALARIU NUMBER(8,2),
FOREIGN KEY (ID_FESTIVAL) REFERENCES FESTIVALURI(ID_FESTIVAL)
);
INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (ID_ANGAJAT 1, ID_FESTIVAL 10, ROL 'logistica', SALARIU 4500.00);
INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (ID_ANGAJAT 2, ID_FESTIVAL 10, ROL 'sunetist', SALARIU 3200.00);
INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (ID_ANGAJAT 3, ID_FESTIVAL 10, ROL 'lumini', SALARIU 5000.00);
INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (ID_ANGAJAT 4, ID_FESTIVAL 11, ROL 'lumini', SALARIU 4800.00);
Output
ANGAJATI
ID_ANGAJAT INT PRIMARY KEY,
ID_FESTIVAL INT,
ROL VARCHAR2(30),
SALARIU NUMBER(8,2),
FOREIGN KEY (ID_FESTIVAL) REFERENCES FESTIVALURI(ID_FESTIVAL)
)
[2025-07-08 19:43:18] completed in 10 ms
C##ROBERT> INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (1, 10, 'logistica', 4500.00)
[2025-07-08 19:43:18] 1 row affected in 5 ms
C##ROBERT> INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (2, 10, 'sunetist', 3200.00)
[2025-07-08 19:43:18] 1 row affected in 2 ms
C##ROBERT> INSERT INTO ANGAJATI (ID_ANGAJAT, ID_FESTIVAL, ROL, SALARIU)
VALUES (3, 10, 'lumini', 5000.00)
[2025-07-08 19:43:18] 1 row affected in 3 ms
Output
ANGAJATI
7 rows
ID_ANGAJAT ID_FESTIVAL ROL SALARIU
1 1 10 logistica 4500.00
2 2 10 sunetist 3200.00
3 3 10 lumini 5000.00
4 4 11 lumini 4800.00
5 5 11 sunet 3000.00
6 6 12 logistica 3900.00
7 7 12 sunet 4100.00

11.Tabela ANGAJATI_CONCERTE

Cod SQL:

```
CREATE TABLE ANGAJATI_CONCERTE (
  ID_ANGAJAT INT,
  ID_CONCERT INT,
  PRIMARY KEY (ID_ANGAJAT, ID_CONCERT),
  FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT),
  FOREIGN KEY (ID_CONCERT) REFERENCES CONCERTE(ID_CONCERT)
);


INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (1, 1);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (2, 3);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (3, 1);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (3, 2);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (4, 4);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (5, 5);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (5, 6);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (6, 6);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (6, 7);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (7, 7);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (4, 2);

INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
VALUES (2, 1);
SELECT *
FROM ANGAJATI_CONCERTE;
```

```
Tx: Manual                                                    Playground
1  CREATE TABLE ANGAJATI_CONCERTE (
2    ID_ANGAJAT INT,
3    ID_CONCERT INT,
4    PRIMARY KEY (ID_ANGAJAT, ID_CONCERT),
5    FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT),
6    FOREIGN KEY (ID_CONCERT) REFERENCES CONCERTE(ID_CONCERT)
7  );
8
9
10 INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
11 VALUES ( ID_ANGAJAT 1, ID_CONCERT 1);
12
13 INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
14 VALUES ( ID_ANGAJAT 2, ID_CONCERT 3);
15
16 INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
17 VALUES ( ID_ANGAJAT 3, ID_CONCERT 1);
18
19 INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
20 VALUES ( ID_ANGAJAT 3, ID_CONCERT 2);
```

Output | ANGAJATI_CONCERTE

```
            ID_ANGAJAT INT,
            ID_CONCERT INT,
            PRIMARY KEY (ID_ANGAJAT, ID_CONCERT),
            FOREIGN KEY (ID_ANGAJAT) REFERENCES ANGAJATI(ID_ANGAJAT),
            FOREIGN KEY (ID_CONCERT) REFERENCES CONCERTE(ID_CONCERT)
          )
[2025-07-08 19:54:24] completed in 9 ms
C##ROBERT> INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
           VALUES (1, 1)
[2025-07-08 19:54:24] 1 row affected in 5 ms
C##ROBERT> INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
           VALUES (2, 3)
[2025-07-08 19:54:24] 1 row affected in 2 ms
C##ROBERT> INSERT INTO ANGAJATI_CONCERTE (ID_ANGAJAT, ID_CONCERT)
           VALUES (3, 1)
[2025-07-08 19:54:24] 1 row affected in 2 ms
```

Output | ANGAJATI_CONCERTE

12 rows

	ID_ANGAJAT	ID_CONCERT
1	1	1
2	2	1
3	2	3
4	3	1
5	3	2
6	4	2
7	4	4
8	5	5
9	5	6
10	6	6
11	6	7
12	7	7

12.Formulati in limbaj natural si implementati 5 cereri SQL complexe

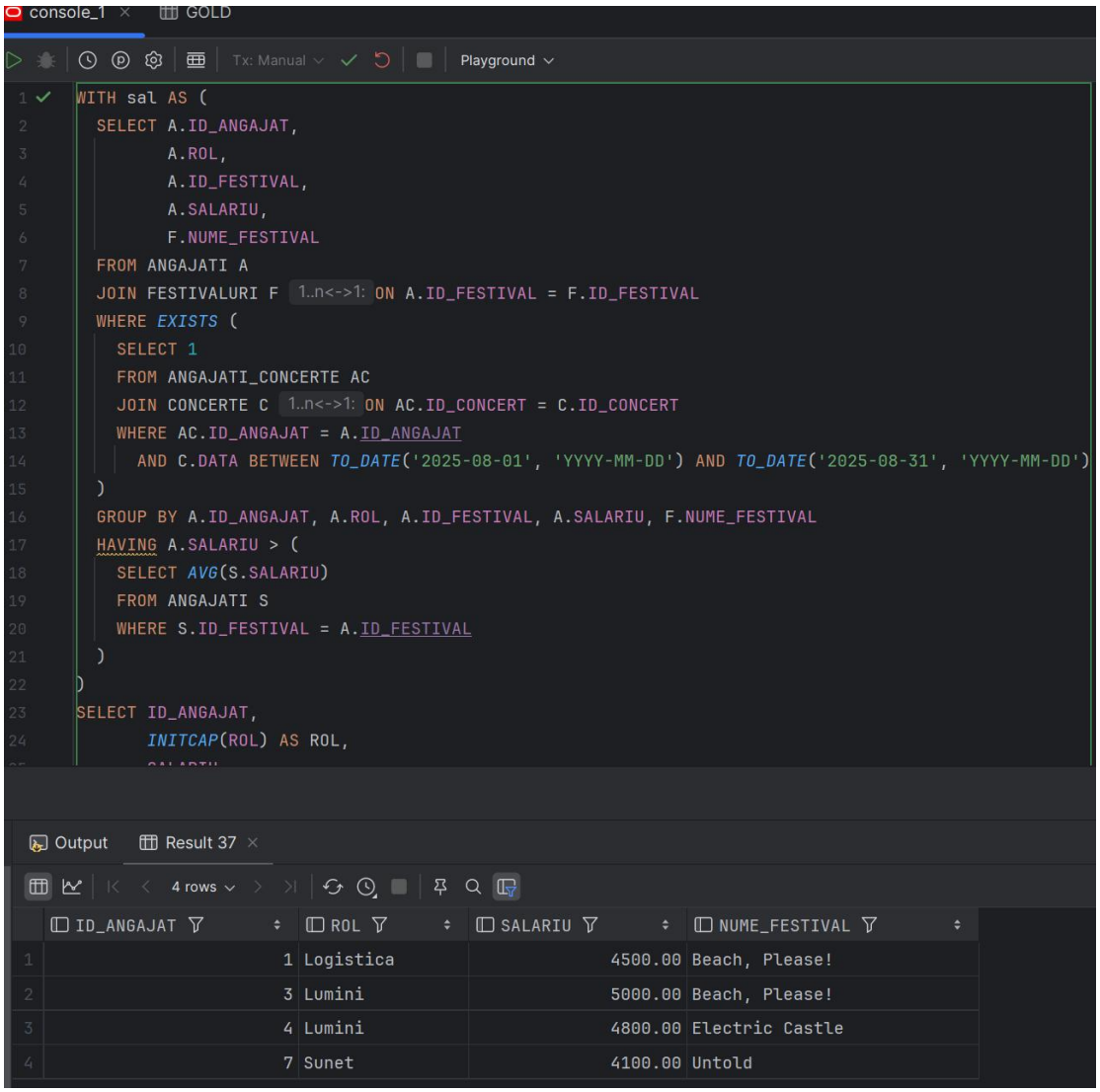
1. Sa se obtina ID-ul si rolul fiecarui angajat care a participat la cel putin un concert in luna august 2025 si are un salariu mai mare decat salariul mediu al angajatilor din acelasi festival. Se va afisa si salariul angajatului, precum si numele festivalului .

Am folosit: subcerere sincronizata, grupari de date, bloc de cerere with si functie de data calendaristica.

Cod SQL:

```
WITH sal AS (
  SELECT A.ID_ANGAJAT,
         A.ROL,
         A.ID_FESTIVAL,
         A.SALARIU,
         F.NUME_FESTIVAL
  FROM ANGAJATI A
  JOIN FESTIVALURI F ON A.ID_FESTIVAL = F.ID_FESTIVAL
  WHERE EXISTS (
    SELECT 1
    FROM ANGAJATI_CONCERTE AC
    JOIN CONCERTE C ON AC.ID_CONCERT = C.ID_CONCERT
    WHERE AC.ID_ANGAJAT = A.ID_ANGAJAT
      AND C.DATA BETWEEN TO_DATE('2025-08-01', 'YYYY-MM-DD') AND
TO_DATE('2025-08-31', 'YYYY-MM-DD')
  )
  GROUP BY A.ID_ANGAJAT, A.ROL, A.ID_FESTIVAL, A.SALARIU,
F.NUME_FESTIVAL
  HAVING A.SALARIU > (
```

```
   SELECT AVG(S.SALARIU)
   FROM ANGAJATI S
   WHERE S.ID_FESTIVAL = A.ID_FESTIVAL
  )
)
SELECT ID_ANGAJAT,
      INITCAP(ROL) AS ROL,
      SALARIU,
      INITCAP(NUME_FESTIVAL) AS NUME_FESTIVAL
FROM sal;
```

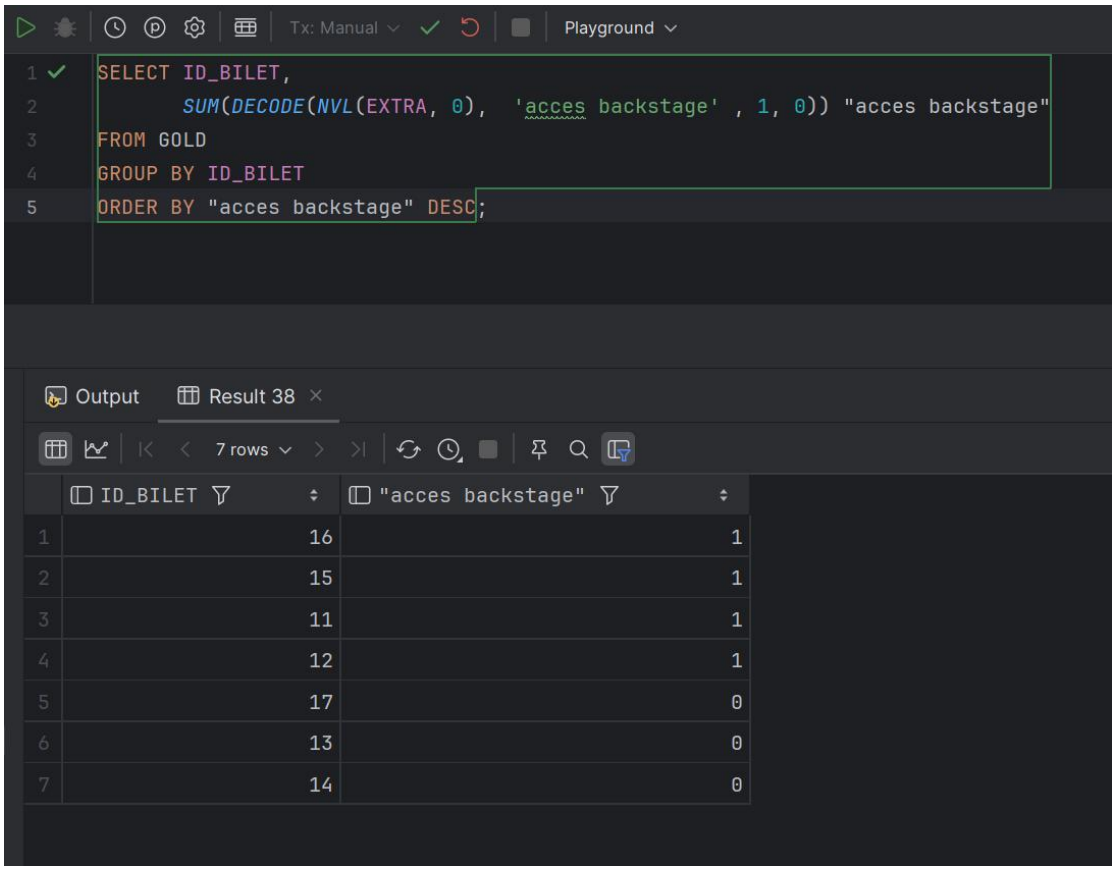


2.Sa se afiseze pentru fiecare bilet gold daca are “ acces backstage” sau nu.

Am folosit: NVL, DECODE si order by.

Cod SQL:

```
SELECT ID_BILET,
      SUM(DECODE(NVL(EXTRA, 0),  'acces backstage' , 1, 0)) "acces backstage"
FROM GOLD
GROUP BY ID_BILET
ORDER BY "acces backstage" DESC;
```

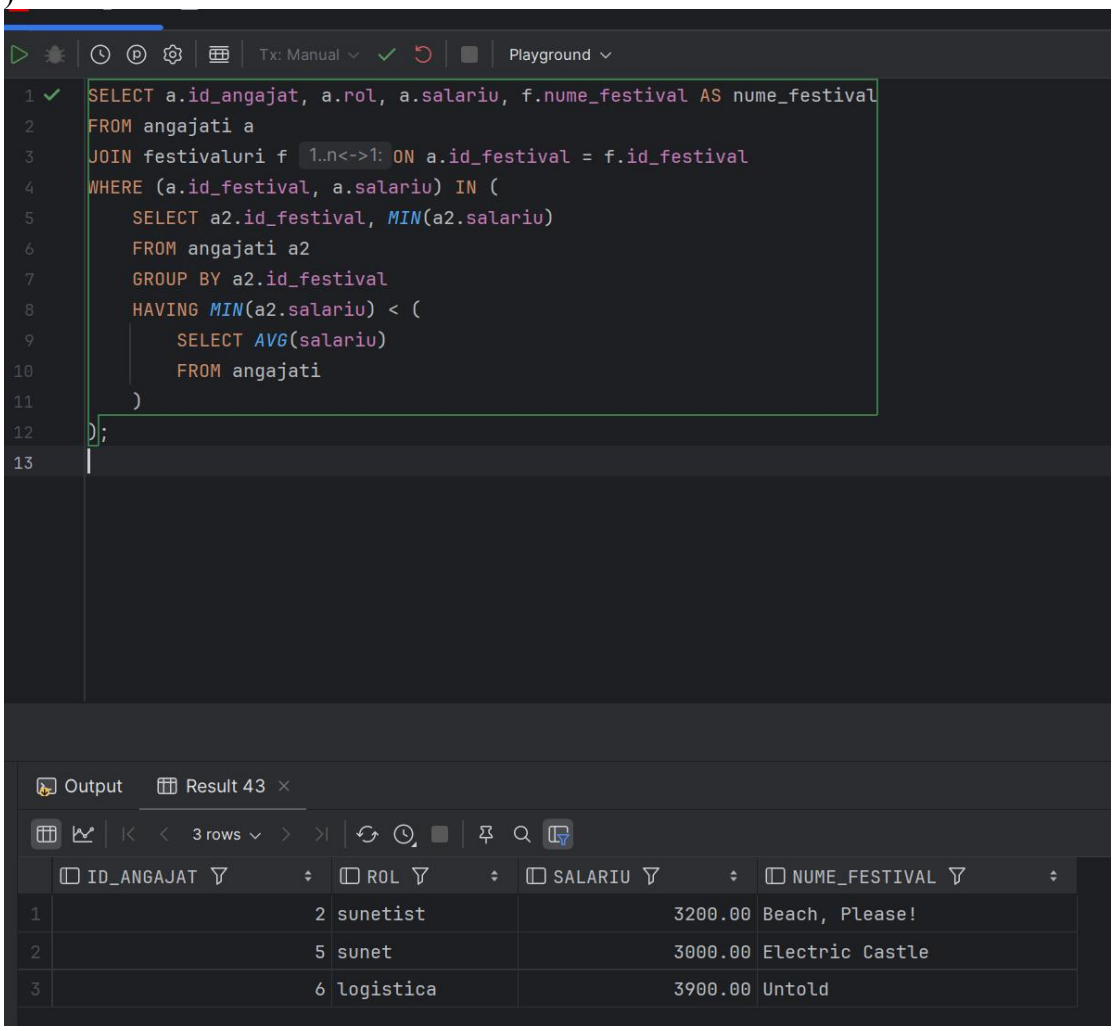


3.Sa se afiseze ID-ul, rolul si salariul celui mai prost platit angajat din fiecare festival, doar daca salariul minim din acel festival este mai mic decat salariul mediu al tuturor angajatilor.

Am folosit: group by, functii de grup (MIN), filtrare la nivel de grupuri cu HAVING, si o subcerere nesincronizata.

```
SELECT a.id_angajat, a.rol, a.salariu, f.nume AS nume_festival
FROM angajati a
JOIN festivaluri f ON a.id_festival = f.id_festival
```

```
WHERE (a.id_festival, a.salariu) IN (
   SELECT a2.id_festival, MIN(a2.salariu)
   FROM angajati a2
   GROUP BY a2.id_festival
   HAVING MIN(a2.salariu) < (
      SELECT AVG(salariu)
      FROM angajati
   )
)
```



4. Sa se afiseze pentru fiecare bilet data achizitiei si:

-daca data este mai recenta de o lună, sa se afiseze mesajul “Mai putin de o luna”;

-daca este intre 1 si 3 luni, sa se afiseze “intre 1 si 3 luni”;

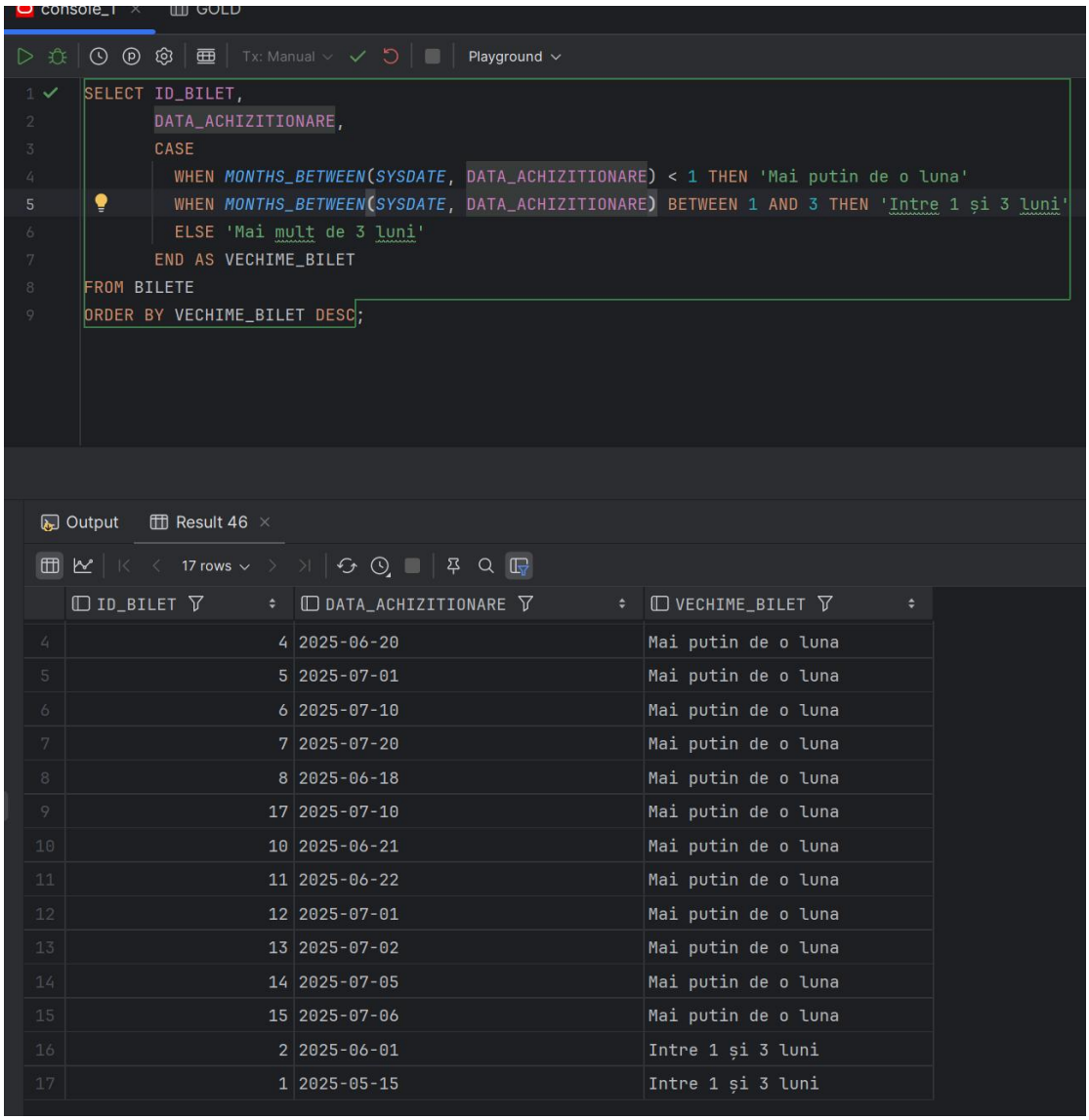
-daca este mai veche de 3 luni, sa se afiseze “Mai mult de 3 luni”.

Am folosit: functii pe date calendaristice: SYSDATE, MONTHS_BETWEEN,

si expresie CASE pentru interpretarea vechimii

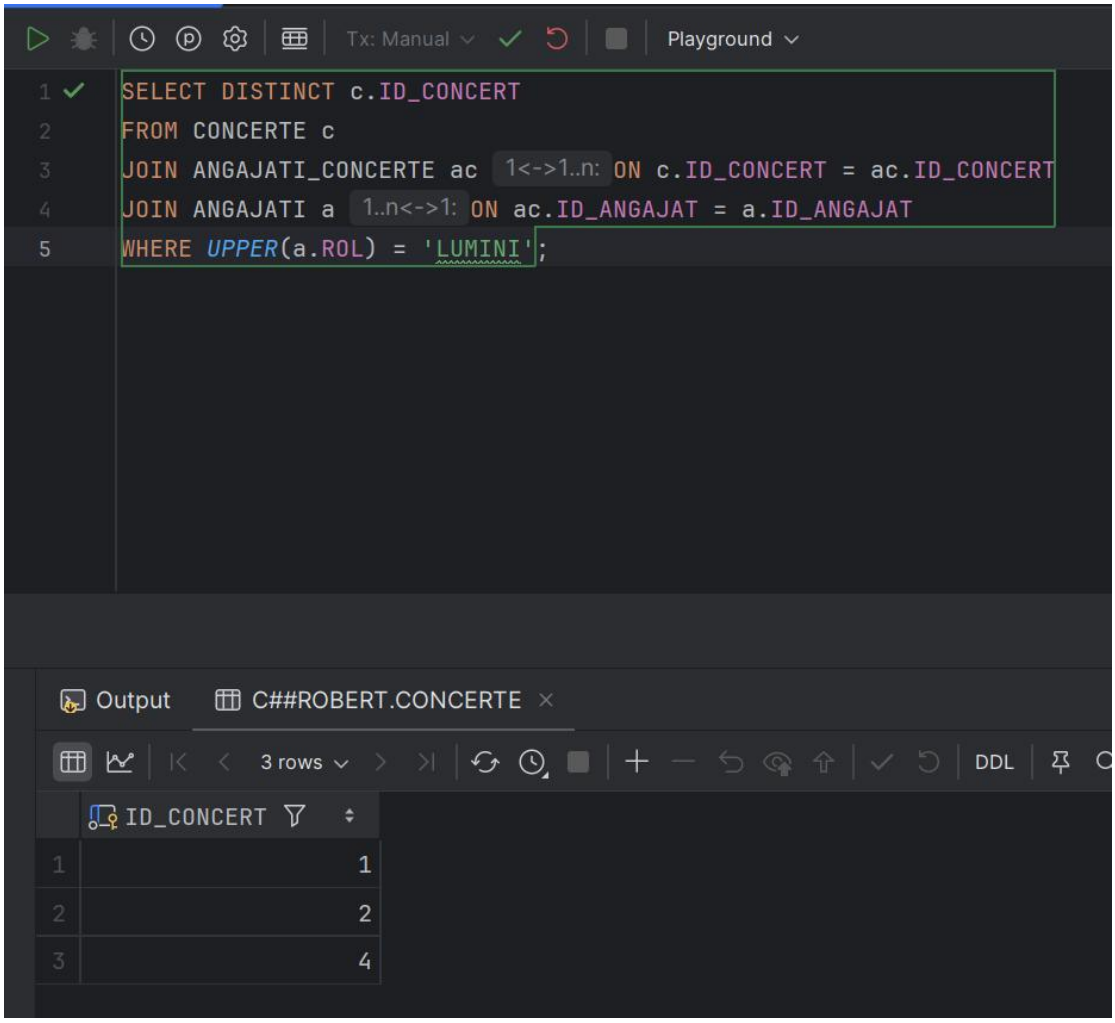
Cod SQL:

```
SELECT ID_BILET,
      DATA_ACHIZITIONARE,
      CASE
        WHEN MONTHS_BETWEEN(SYSDATE, DATA_ACHIZITIONARE) < 1
THEN 'Mai putin de o luna'
        WHEN MONTHS_BETWEEN(SYSDATE, DATA_ACHIZITIONARE)
BETWEEN 1 AND 3 THEN 'Intre 1 și 3 luni'
        ELSE 'Mai mult de 3 luni'
      END AS VECHIME_BILET
FROM BILETE
ORDER BY VECHIME_BILET DESC;
```



5.Sa se afiseze id-ul concertelor la care au lucrat persoane cu rolul "lumini".
Am folosit: subcereri sincronizate si functie pe siruri de caractere (UPPER).

```
SELECT DISTINCT c.ID_CONCERT
FROM CONCERTE c
JOIN ANGAJATI_CONCERTE ac ON c.ID_CONCERT = ac.ID_CONCERT
JOIN ANGAJATI a ON ac.ID_ANGAJAT = a.ID_ANGAJAT
WHERE UPPER(a.ROL) = 'LUMINI';
```

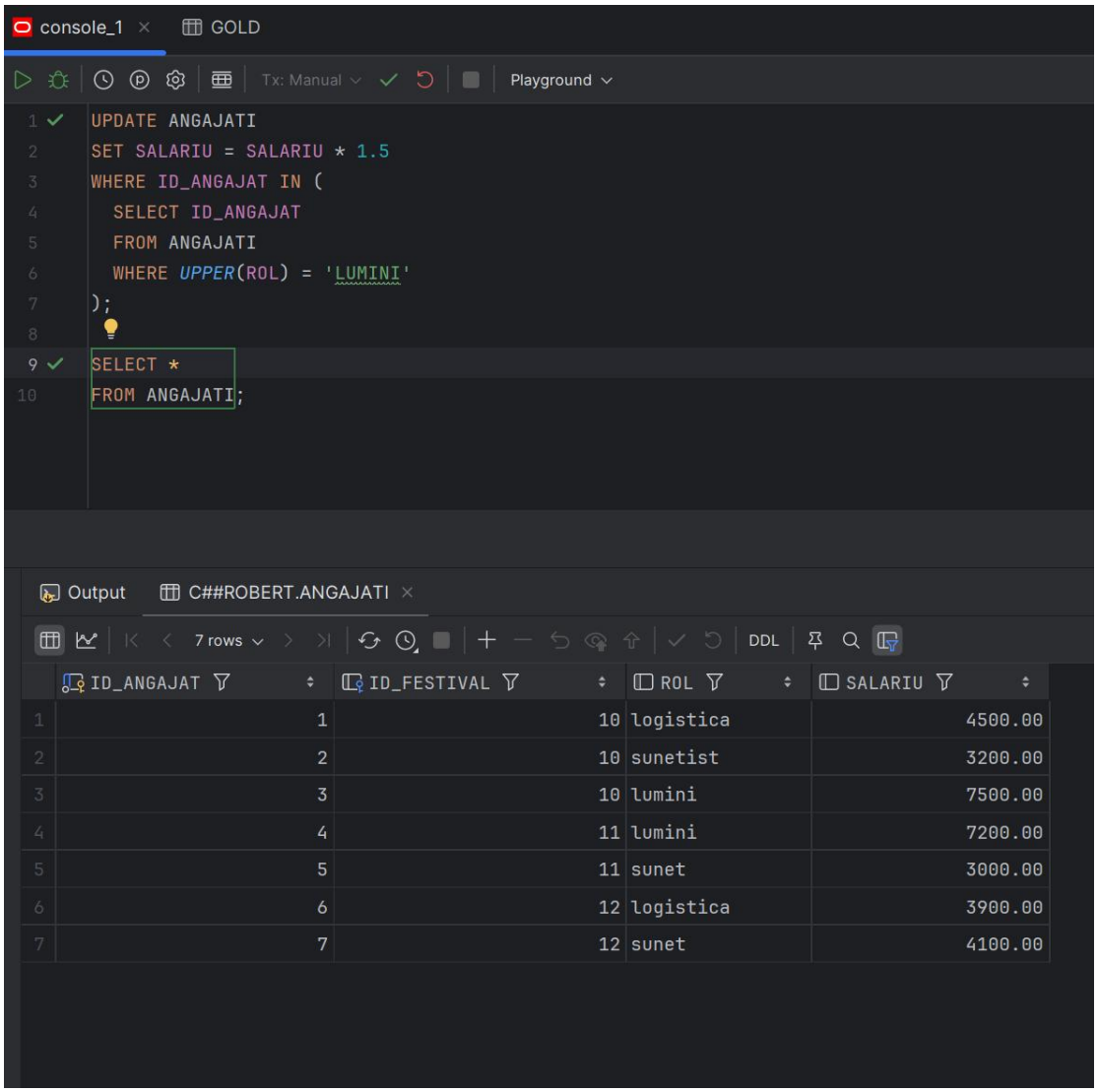


13.Implementarea a 3 operatii de actualizare si de suprimare a datelor utilizand subcereri

1.Sa se creasca cu 50% salariul angajatilor care au rolul „lumini”.

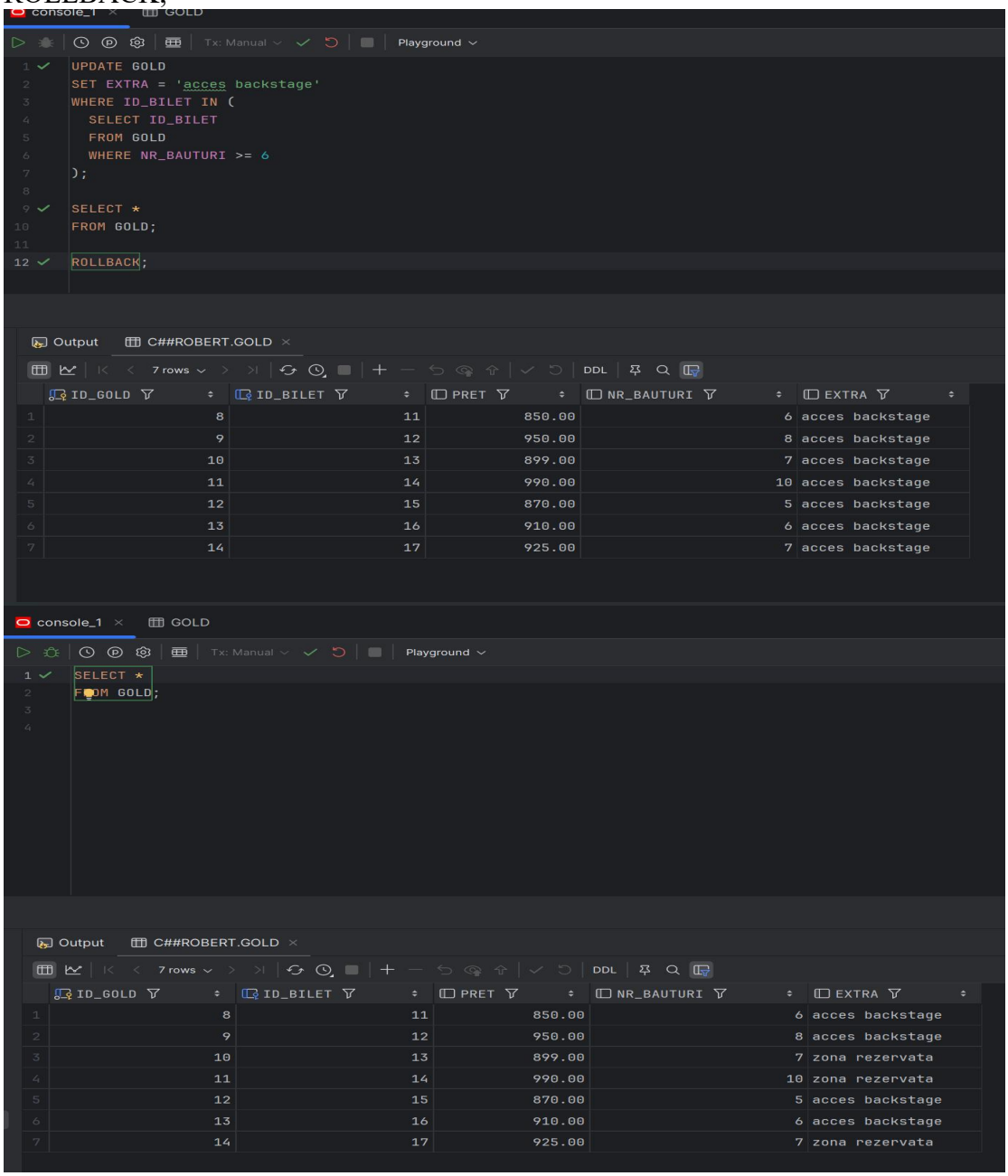
Codul SQL:

```
UPDATE ANGAJATI
SET SALARIU = SALARIU * 1.5
WHERE ID_ANGAJAT IN (
  SELECT ID_ANGAJAT
  FROM ANGAJATI
  WHERE UPPER(ROL) = 'LUMINI'
);

SELECT *
FROM ANGAJATI;
```



2.Sa se modifice tabela GOLD astfel incat in coloana EXTRA toti cei care au mai mult de 6 bauturi sa aiba „acces backstage”

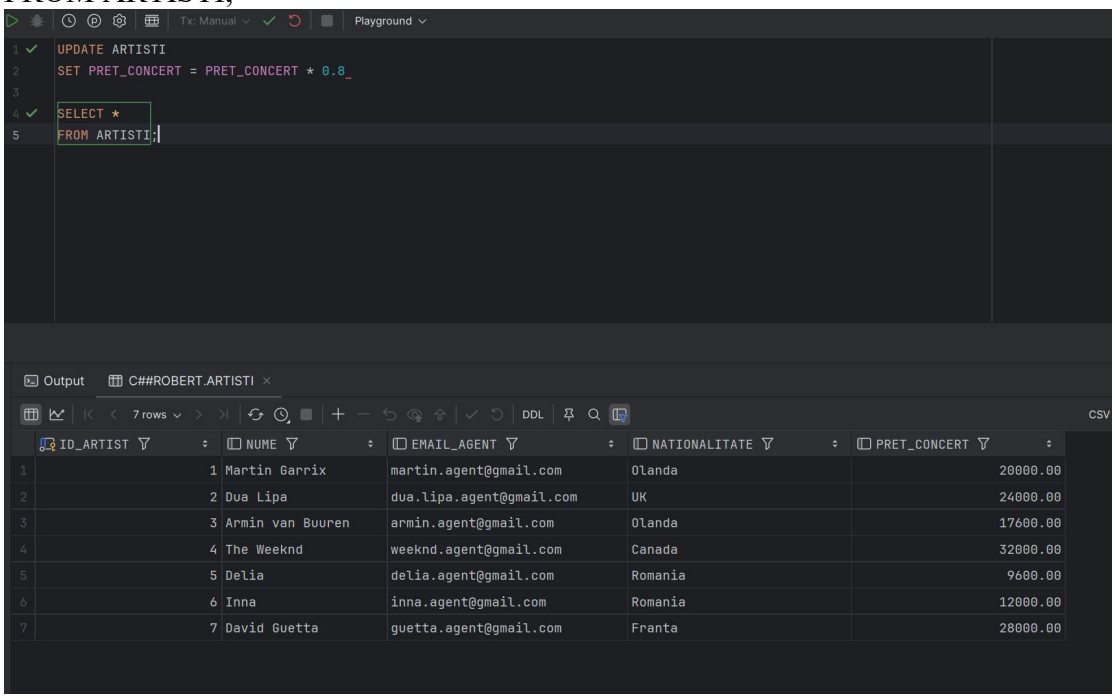
```
UPDATE GOLD
SET EXTRA = 'acces backstage'
WHERE ID_BILET IN (
  SELECT ID_BILET
  FROM GOLD
  WHERE NR_BAUTURI >= 6
);

SELECT *
FROM GOLD;

ROLLBACK;
```



3.Sa se scada cu 20% pretul tuturor artistilor.

Cod SQL:

```
UPDATE ARTISTI
SET PRET_CONCERT = PRET_CONCERT * 0.8

SELECT *
FROM ARTISTI;
```



	ID_ARTIST	NUME	EMAIL_AGENT	NATIONALITATE	PRET_CONCERT
1	1	Martin Garrix	martin.agent@gmail.com	Olanda	20000.00
2	2	Dua Lipa	dua.lipa.agent@gmail.com	UK	24000.00
3	3	Armin van Buuren	armin.agent@gmail.com	Olanda	17600.00
4	4	The Weeknd	weeknd.agent@gmail.com	Canada	32000.00
5	5	Delia	delia.agent@gmail.com	Romania	9600.00
6	6	Inna	inna.agent@gmail.com	Romania	12000.00
7	7	David Guetta	guetta.agent@gmail.com	Franta	28000.00