

## ***PROIECT - BAZE DE DATE***

Cerințe:

### **1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.**

Baza de date pe care o voi crea definește un sistem de monitorizare a unor săli fitness din diferite lanțuri de firme. Presupun că am o aplicație online ce îi permite utilizatorului să își deschidă un abonament la o sală din oricare oraș. Aceste abonamente pot fi de mai multe tipuri, în funcție de perioada aleasă (săptămânal, lunar sau anual) sau de clasele la care s-a înscris, fapt pentru care trebuie contorizate plățile abonamentelor. Fiecare sală fitness are mai mulți traineri pe diferite filiere. Aceștia pot fi traineri personali sau pot susține clase private (zumba, pilates, yoga, etc). Astfel, utilizatorul poate alege la ce clase dorește să participe făcându-și o rezervare pentru o anumită dată. Tot din aplicație, utilizatorul poate vedea în timp real pentru fiecare sală fitness echipamentul de care dispune, dar și diferite date despre mențenanța acestuia, cât și angajații.

### **2. Prezentarea constrângerilor (restrictii, reguli) impuse asupra modelului.**

O sală fitness poate avea mai mulți membrii (abonați), mai multe abonamente, mai multe clase, mai mulți traineri, mai multe echipamente și mai mulți angajați.

Un membru (abonat) își poate cumpăra abonament la o singură sală fitness, își poate face mai multe rezervări la clase diferite (nu își poate face rezervare de mai multe ori la aceeași clasă, dar poate alege clase diferite, sau într-un alt interval orar). Acesta poate plăti o singură dată un abonament.

Un tip de abonament poate fi comun la mai mulți membrii, dar poate fi efectuată doar o plată pentru acesta.

O plată de abonament poate fi efectuată doar pentru un singur abonament și pentru un singur membru.

Un trainer poate să-și desfășoare activitatea la o singură sală fitness și poate să susțină mai multe clase private, având astfel sub coordonare mai mulți membrii cu abonamente sau rezervări.

O clasă privată poate avea un singur trainer ca responsabil, dar se poate regăsi același tip de clasă la mai multe săli diferite.

O rezervare pentru o clasă poate fi făcută doar o singură dată de un membru.

Același tip de echipament se poate regăsi la mai multe săli fitness, iar pentru acesta se pot realiza mai multe verificări de mențenanță, dar la date diferite.

Un angajat poate să lucreze doar la o singură sală fitness pe un anumit post.

### 3. Descrierea entităților, inclusiv precizarea cheii primare.

ENTITATE	CHEIE PRIMARĂ	OBSERVAȚII
sali_fitness	id_sala	Informații generale despre fiecare sală fitness.
membrii	id_membru	Informații despre fiecare utilizator, dar și sala fitness specifică la care s-a înscris.
abonamente	id_abonament	Date despre abonamentul ales de un anumit membru la o sală fitness.
plati_abonamente	id_abonament	Subentitate a entității abonamente, reține informațiile necesare plății unui abonament de către un membru.
clase	id_clasa	Informații generale despre desfășurarea unei clase private la o anumită sală.
traineri	id_trainer	Reține informații despre trainerii ce-și desfășoară serviciile la o sală fitness.
echipamente	id_echipament	Informații generale despre echipamentele ce aparțin fiecărei săli.
mentenanta_echipament	id_echipament	Subentitate a entității echipamente, monitorizează informațiile despre ultima verificare de menenanță a unui tip de echipament.
angajati	id_angajat	Informații despre fiecare angajat al unei săli fitness.
rezervari	id_rezervare	Informații despre rezervările unui membru la clasele private desfășurate într-o anumită sală fitness.

### 4. Descrierea relațiilor, inclusiv precizarea cardinalității acestora.

RELAȚIE	CARDINALITATE	OBSERVAȚII
lucrează	sali_fitness - angajat one-to-many	O sală fitness poate avea mai mulți angajați, doar că un singur angajat poate lucra doar la o sală fitness.
deține	sali_fitness - echipament one-to-many	O sală fitness poate deține mai multe echipamente.

reține	sali_fitness - (membru, clasa, trainer, abonament) one-to-many	O sală fitness reține mai mulți membrii, mai multe clase, mai multe abonamente și mai mulți traineri. Doar că un membru, un trainer, o clasă și un abonament specific pot apartine doar de o anumită sală fitness.
antrenează	trainer - membru one-to-many	Un trainer poate fi antrenor personal pentru mai mulți membrii, doar că un membru poate fi antrenat de un singur trainer.
folosește	membru - echipament many-to-many	Mai mulți membrii pot folosi mai multe echipamente din sala fitness.
cumpără	abonament - membru one-to-many	Un membru poate achiziționa un singur abonament la o sală anume.
participă	trainer - clasa one-to-many membru - clasa many-to-many	Un singur trainer poate coordona o clasă, dar poate susține mai multe clase diferite. Mai mulți membrii pot participa la mai multe clase.
rezervă	rezervare - membru one-to-many	O rezervare poate fi făcută de mai mulți membrii, dar un membru nu poate efectua mai mult de o rezervare pentru o clasă.
efectuată pentru	rezervare - clase many-to-many	Mai multe rezervări pot fi efectuate pentru mai multe clase, iar mai multe clase pot avea mai multe rezervări.

## 5. Descrierea atributelor, incluzând tipul de date și eventuale constrângeri, valori implicate, valori posibile ale atributelor.

ENTITATE: SALI\_FITNESS

Atribut	Tip de date	Dimensiunea/ Precizie	Valori posibile și valori default	Observații, obligatoriu/optional
id_sala	INT	PK	Fiecare sală fitness are un id unic.	Este obligatoriu.
locatie	VARCHAR2(50)	NOT NULL	Reține locația fiecărei săli	Obligatoriu, NOT NULL.

			fitness.	
lant_firma	VARCHAR2(50)	NOT NULL	Reține lanțul de firme din care face parte sala fitness.	Obligatoriu.
nr_abonati	INT		Reține numărul abonaților de la fiecare sala fitness.	Opțional.
nr_traineri	INT		Reține numărul de traineri de la fiecare sală fitness.	Opțional.
nr_echipamente	INT		Reține numărul de echipamente din fiecare sală fitness.	Opțional.
nr_telefon	VARCHAR2(10)	10	Reține numărul de telefon al unei săli fitness și este unic.	Opțional.
ora_deschidere	DATE		Reține ora la care se deschide sala fitness.	Opțional.
ora_inchidere	DATE		Reține ora la care se închide sala fitness.	Opțional.

#### ENTITATE: MEMBRU

Atribut	Tip de date	Dimensiunea/ Precizie	Valori posibile și valori default	Observații, obligatoriu/opțional
id_membru	INT	PK	Fiecare membru are un id unic.	Obligatoriu.
id_sala	INT	FK	Fiecare membru din sală îi corespunde un id unic de care aparține.	Obligatoriu, NOT NULL.

id_trainer	INT	FK	Fiecărui membru din sală îi corespunde un singur id_trainer, reprezentând trainer-ul de care este antrenat.	Obligatoriu. NOT NULL.
nume	VARCHAR2(50)	NOT NULL	Reține numele membrului ce și-a făcut abonamentul la sală.	Obligatoriu.
email	VARCHAR2(50)		Reține adresa de email al membrului și este unică.	Optional.
nr_telefon	VARCHAR2(10)	10	Reține numărul de telefon al membrului și este unic.	Obligatoriu. NOT NULL.
data_inscriere	DATE	DEFAULT SYSDATE	Reține data la care membrul s-a înscris pe platformă.	Optional.

#### ENTITATE: ABONAMENT

Atribut	Tip de date	Dimensiune/ Precizie	Valori posibile și valori default	Observații, obligatoriu / optional
id_abonament	INT	PK	Reține id-ul unic al unui abonament efectuat pe platformă.	Obligatoriu.
id_membru	INT	FK	Fiecărui abonament de membru îi corespunde id-ul unic al membrului.	Obligatoriu, NOT NULL.
id_sala	INT	FK	Fiecărui abonament îi corespunde id-ul unic al sălii de care aparține.	Obligatoriu, NOT NULL.
tip_abonament	VARCHAR2(50)	NOT NULL	săptămânal / lunar / anul / 10 Ședințe (reține tipul abonamentului fiecărui membru).	Obligatoriu, NOT NULL.
data_start	DATE		Reține data de	Obligatoriu,

			începere a abonamentului selectat.	NOT NULL.
data_end	DATE		Reține data de sfârșit a abonamentului selectat.	Optional.
pret	NUMBER(10, 2)		Reține prețul abonamentului în funcție de tipul acestuia.	Optional.

#### ENTITATE: PLATI\_ABONAMENTE

Atribut	Tip de date	Dimensiune/ Precizie	Valori posibile și valori default	Observații, obligatoriu / optional
id_membru	INT	FK	Fiecărei plăți îi corespunde id-ul unic al membrului.	Obligatoriu, NOT NULL.
id_abonament	INT	PK	Fiecărei plăți îi corespunde ca primary key id-ul unic al abonamentului.	Obligatoriu, NOT NULL.
suma	NUMBER(10, 2)		Suma plătită de membru pentru abonamentul selectat.	Optional.
data_plata	DATE	DEFAULT SYSDATE	Data la care a fost efectuată plata.	Optional.
metoda_plata	VARCHAR(50)		card / transfer bancar / cash (metoda prin care a fost efectuată plata).	Optional.
status_plata	VARCHAR2(50)		efectuată / eșuată / în curs	Optional

### ENTITATE: CLASA

Atribut	Tip de date	Dimensiune / Precizie	Valori posibile și valori default	Observații, obligatoriu / optional
id_clasa	INT	PK	Fiecărei clase îi corespunde un id-unic.	Obligatoriu.
id_sala	INT	FK	Fiecărei clase îi corespunde id-ul sălii de care aparține.	Obligatoriu.
id_trainer	INT	FK	Fiecărei clase îi corespunde id-ul trainerului ce va susține clasa respectivă.	Obligatoriu.
nume	VARCHAR2(100)	NOT NULL	Reține numele clasei ce se va desfășura.	Obligatoriu.
capacitate_maxima	INT		Reține numărul de membrii maxim care pot participa la clasa respectivă.	Optional.

### ENTITATE: REZERVĂRI

Atribut	Tip de date	Dimensiune / Precizie	Valori posibile și valori default	Observații, optional / obligatoriu
id_rezervare	INT	PK	Fiecărei rezervări îi corespunde un id unic ca primary key.	Obligatoriu.
id_membru	INT	FK	Fiecărei rezervări îi corespunde id-ul unic al membrului căruia îi aparține.	Obligatoriu.
id_clasa	INT	FK	Fiecărei	Obligatoriu.

			rezervări îi corespunde id-ul unic al clasei de care aparține rezervarea.	
data_rezervare	DATE	NOT NULL	Reține data pentru care a fost făcută rezervarea.	Obligatoriu.
status	VARCHAR2(50)	NOT NULL	confirmată / neconfirmată	Opțional

#### ENTITATE: TRAINER

Atribut	Tip de date	Dimensiune / Precizie	Valori posibile și valori default	Observații, obligatoriu / opțional
id_trainer	INT	PK	Fie căruiai trainer îi corespunde un id unic ca primary key.	Obligatoriu.
id_sala	INT	FK	Fie căruiai trainer îi corespunde id-ul sălii de care aparține.	Obligatoriu.
nume	VARCHAR2(50)	NOT NULL	Reține numele trainerului, este unic.	Obligatoriu.
nr_telefon	NUMBER(10)	NOT NULL	Reține numărul de telefon al trainerului, este unic.	Obligatoriu.
email	VARCHAR2(50)	NOT NULL	Reține adresa de email al trainerului, este unic.	Opțional.
specializare	VARCHAR2(50)	NOT NULL	Reține specializarea trainerului angajat la sala	Obligatoriu.

			respectivă.	
salariu	NUMBER(15, 2)		Reține salariul trainerului angajat la sala respectivă.	Optional.

#### ENTITATE: ANGAJAT

Atribut	Tip de date	Dimensiune / Precizie	Valori posibile și valori default	Observații, obligatoriu / optional
id_angajat	INT	PK	Fiecare angajat are un id unic ca primary key.	Obligatoriu.
id_sala	INT	FK	Fie căruia angajat îi corespunde id-ul unic al sălii de care aparține.	Obligatoriu.
nume	VARCHAR2(50)	NOT NULL	Reține numele angajatului ce lucrează la sala respectivă, este unic.	Obligatoriu.
functie	VARCHAR2(30)	NOT NULL	Reține funcția specifică fie căruia angajat.	Obligatoriu.
salariu	NUMBER(15, 2)	NOT NULL	Reține salariul fie căruia angajat al unei săli fitness.	Optional.
telefon	VARCHAR2(10)	NOT NULL	Reține numărul de telefon al unui angajat, este unic.	Obligatoriu.
email	VARCHAR2(50)	NOT NULL	Reține adresa de email al fie căruia angajat, este unic.	Obligatoriu.

**ENTITATE: ECHIPAMENT**

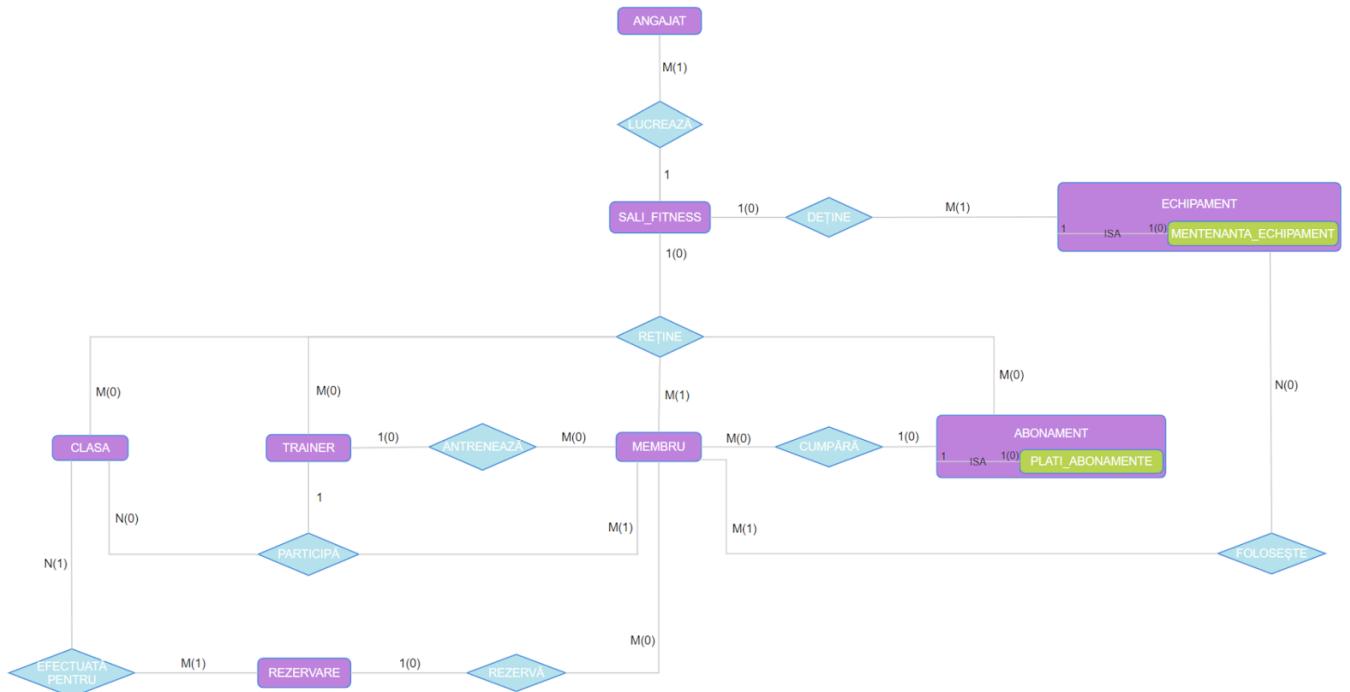
Atribut	Tip de date	Dimensiune / Precizie	Valori posibile și valori default	Observații, optional / obligatoriu
id_echipament	INT	PK	Fiecare echipament are un id unic ca primary key.	Obligatoriu.
id_sala	INT	FK	Fie căru echipament îi corespunde id-ul unic al sălii de care aparține.	Obligatoriu.
denumire	VARCHAR2(50)	NOT NULL	Reține numele specific al unui echipament.	Obligatoriu.
tip_abonament	VARCHAR2(50)		Reține tipul din care face parte echipamentul respectiv.	Optional.
status	VARCHAR2(50)		Se rețin anumite comentarii despre echipamentele respective.	Optional.
nr_total	INT		Reține numărul total de echipamente dintr-un anumit tip din sala respectivă.	Optional.

**ENTITATE: MENTENANTA\_ECHIPAMENT**

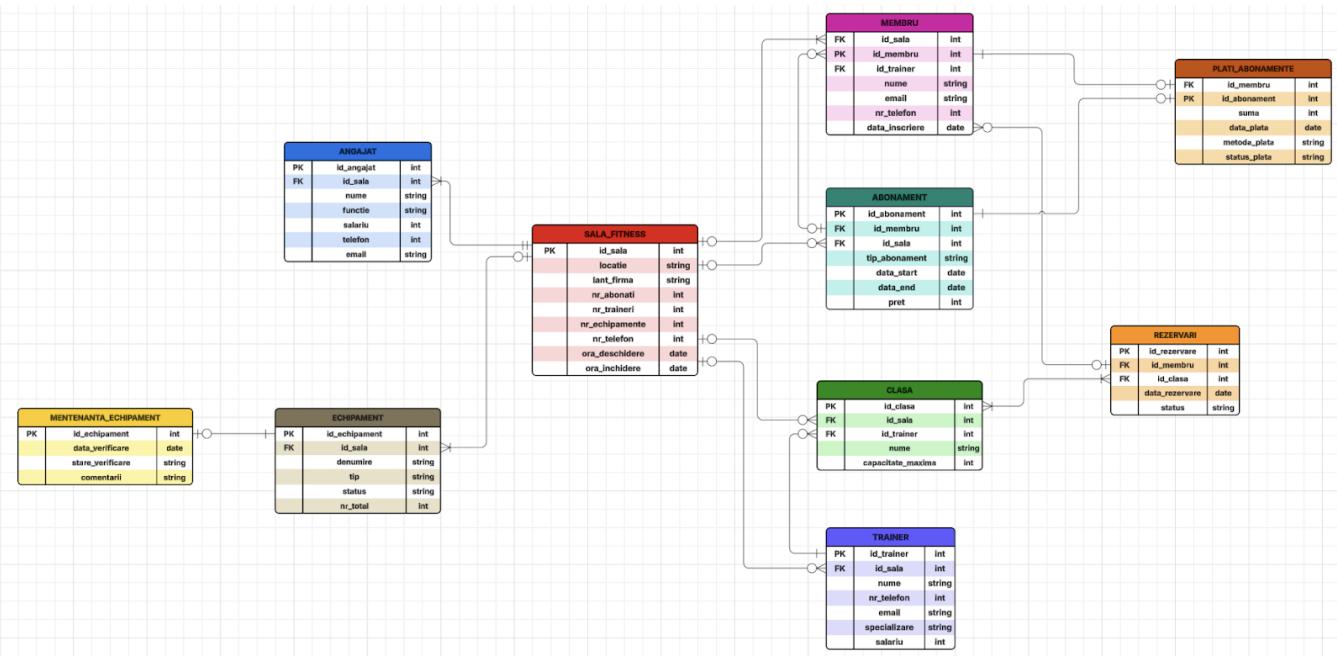
Atribut	Tip de date	Dimensiune / Precizie	Valori posibile și valori default	Observații, optional / obligatoriu
id_echipament	INT	PK	Fiecare echipament reține ca	Obligatoriu.

			primary key, id-ul unic al echipamentului.	
data_verificare	DATE	DEFAULT SYSDATE	Reține data ultimei verificări a unui echipament.	Optional.
stare_verificare	VARCHAR2(100)	NOT NULL	funcțional / defect	Obligatoriu.
comentarii	VARCHAR2(500)	NOT NULL	Se rețin comentarii despre verificarea unui echipament.	Optional.

## 6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.



Am mai efectuat o diagramă pentru a reprezenta tabelele specifice proiectului meu:



7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 7 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.

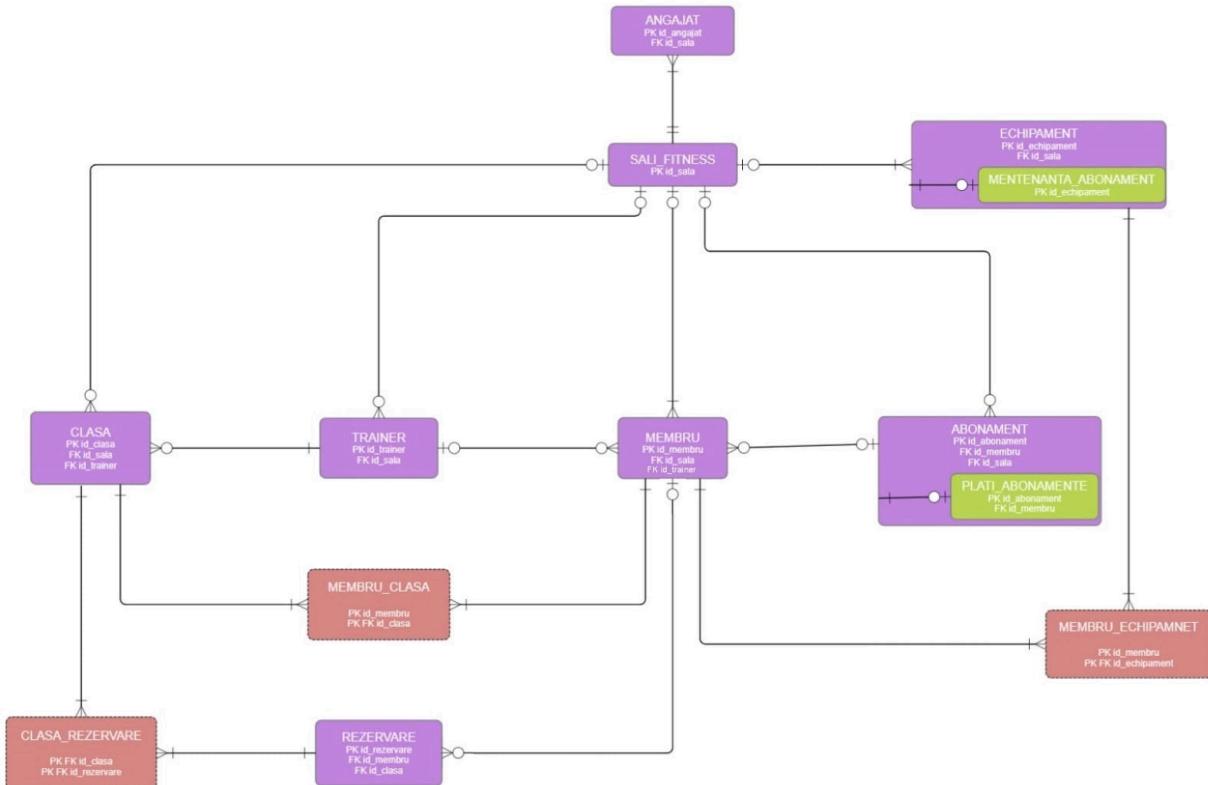
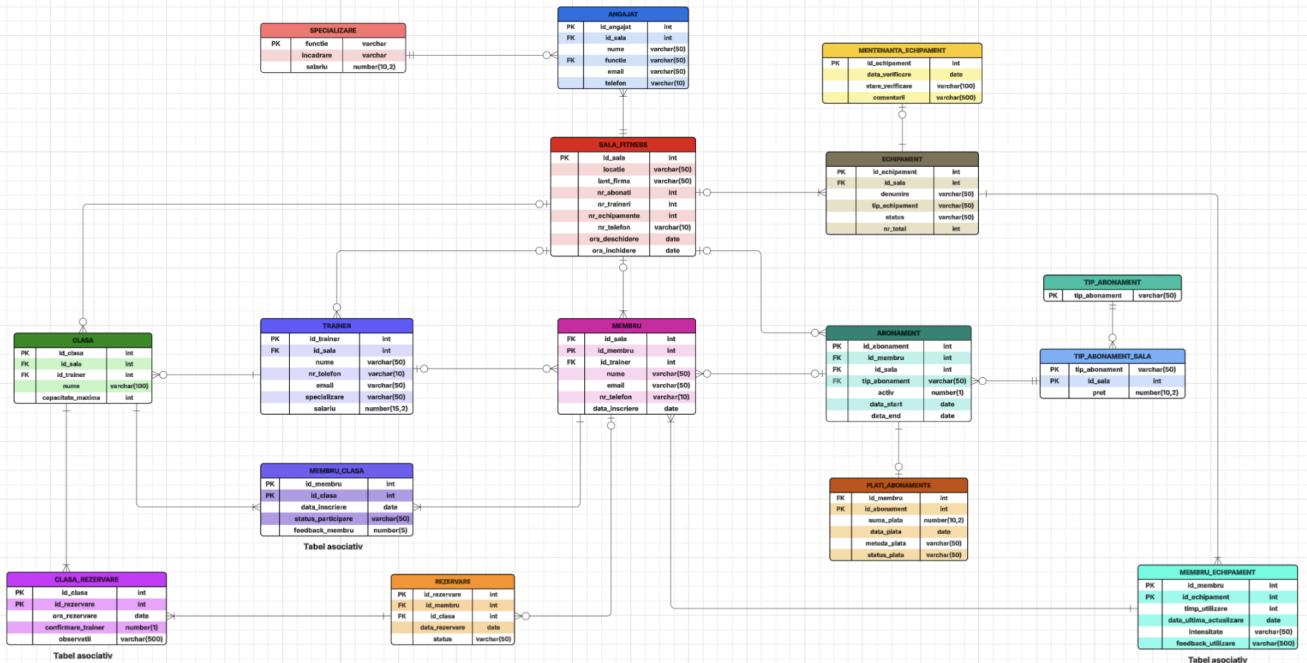


Diagrama conceptuală după normalizarea în FN3:



Continuare cerințe Partea a două:

## 8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

### SCHEME RELATIONALE:

Tabele neasociative:

SALA\_FITNESS(#id\_sala, locatie, lant\_firma, nr\_abonati, nr\_traineri, nr\_echipamente, nr\_telefon, ora\_deschidere\_sala, ora\_inchidere\_sala)

MEMBRU(#id\_membru, #id\_sala, id\_trainer, nume\_membru, email\_membru, nr\_telefon, data\_inscriere)

TIP\_ABONAMENT(#tip\_abonament)

TIP\_ABONAMENT\_SALA(#tip\_abonament, #id\_sala, pret)

ABONAMENT(#id\_abonament, #id\_sala, #id\_membru, tip\_abonament, activ, data\_start, data\_end)

SPECIALIZARE(#functie, incadrare, salariu)

ANGAJAT(#id\_angajat, #id\_sala, nume, functie, telefon, email)

PLATA\_ABONAMENT(#id\_abonament, #id\_membru, suma, data\_plata, metoda\_plata, status\_plata)

CLASA(#id\_clasa, #id\_sala, #id\_trainer, nume\_clasa, capacitate\_maxima)

TRAINER(#id\_trainer, #id\_sala, nume, nr\_telefon, email, specializare, salariu)

REZERVARI(#id\_rezervare, #id\_membru, #id\_clasa, data\_rezervare, status)

ECHIPAMENT(#id\_echipament, #id\_sala, denumire, tip\_echipament, nr\_total)

MENTENANTA\_EQUIPAMENT(#id\_echipament, data\_verificare, stare\_verificare, comentarii)

Tabele asociative:

MEMBRU\_CLASA(#id\_membru, #id\_clasa, data\_inscriere\_clasa, status\_participare, feedback\_membru)

MEMBRU\_ECHIPAMENT(#id\_membru, #id\_echipament, timp\_utilizare, data\_ultima\_utilizare, intensitate, feedback\_utilizare)

CLASA\_REZERVARE(#id\_clasa, #id\_rezervare, ora\_rezervare, confirmare\_trainer, observatii)

## 9. Realizarea normalizării până la forma normală 3 (FN1-FN3).

SALA\_FITNESS(#id\_sala, locatie, lant\_firma, nr\_abonati, nr\_traineri, nr\_echipamente, nr\_telefon, ora\_deschidere, ora\_inchidere) :

- FN1: toate valorile sunt atomice, deci este în FN1.
- FN2: cheia este id\_sala, toate celelalte atribute depind direct de ea, deci este în FN2.
- FN3: nu există dependențe între atributele ne-cheie, aşadar este în FN3.

MEMBRU (#id\_membru, id\_sala, id\_trainer, nume, email, nr\_telefon, data\_inscriere) :

- FN1: toate atributele sunt atomice, deci este în FN1.
- FN2: cheia primară este id\_membru, restul atributelor nu depind complet de ea, deci este în FN2.
- FN3: id\_sala și id\_trainer pot duce la dependențe tranzitive, însă aceste relații sunt deja rezolvate în schemă, prin relații cu tabele separate (TRAINER și SALA\_FITNESS). Așadar, este în FN3.

ABONAMENT (#id\_abonament, id\_membru, id\_sala, tip\_abonament, data\_start, data\_end, pret) :

- FN1: toate atributele sunt atomice, deci este în FN1.
- FN2: cheia este id\_abonament, iar restul atributelor depind direct de ea, deci este în FN2.
- FN3: atributul tip\_abonament este determinat de atributul pret, aşadar avem dependență tranzitivă. Însă, tip\_abonament trebuie să fie legat și de id\_sala (fiecare sală își impune un anumit preț în funcție de tip\_abonament).

REZOLVARE: fac o tabela separată pentru tip\_abonament -> TIP\_ABONAMENT(tip\_abonament).

scot tip\_abonament, pret și id\_sala într-un tabel separat -> TIP\_ABONAMENT\_SALA(tip\_abonament, id\_sala, pret).

Avem următoarele tabele:

```
TIP_ABONAMENT(  
    tip_abonament VARCHAR2(50) PRIMARY KEY  
)  
TIP_ABONAMENT_SALA(  
    tip_abonament VARCHAR2(50),  
    id_sala INT FOREIGN KEY,  
    pret NUMBER(10, 2)
```

```

PRIMARY KEY (tip_abonament, id_sala)
)
ABONAMENT(
    id_abonament INT PRIMARY KEY,
    id_membru INT FOREIGN KEY,
    id_sala INT FOREIGN KEY,
    tip_abonament VARCHAR2(50) FOREIGN KEY,
    data_start DATE,
    data_end DATE
)

```

Toate sunt acum în FN3.

PLATA\_ABONAMENT (#id\_abonament, id\_membru, suma\_plata, data\_plata, metoda\_plata, status\_plata) :

- FN1: toate atributele sunt atomice, este în FN1.
- FN2: cheia este id\_abonament, nu este compusă, aşadar atributele depind complet de ea.  
În schimb, id\_abonament determină deja id\_membru, deci ultimul este reduncant și îl voi elimina.

Pentru a fi în FN2, tabelul devine:

PLATA\_ABONAMENT (id\_abonament, suma, data\_plata, metoda\_plata, status\_plata).

- FN3: nu există dependențe tranzitive între alte atrbute, aşadar este în FN3.

ANGAJAT (#id\_angajat, id\_sala, nume, functie, salariu, telefon, email) :

- FN1: toate atrbutele sunt atomice, este în FN1.
- FN2: cheia este id\_angajat, toate celelalte atrbute depind direct de ea, deci este în FN2.
- FN3: atrbutul salariu este determinat de atrbutul functie, aşadar avem dependență tranzitivă.

REZOLVARE: scot salariu și functie într-un tabel separat -> SPECIALIZARE(salariu, functie).

Avem următoarele tabele:

```

SPECIALIZARE (
    salariu NUMBER(10, 2)
    functie VARCHAR2(50)
)
ANGAJAT (
    id_angajat INT PRIMARY KEY,
    id_sala INT FOREIGN KEY REFERENCE SALA_FITNESS(id_sala),
    nume VARCHAR2(50),
    functie NUMBER(15, 2) FOREIGN KEY,
    telefon NUMBER(10),
    email VARCHAR2(50)
)

```

Ambele sunt acum în FN3.

TRAINER (#id\_trainer, id\_sala, nume, nr\_telefon, email, specializare, salariu) :

- FN1: toate atributele sunt atomice, este în FN1.
- FN2: cheia este id\_trainer, toate celelalte atrbute depind direct de ea, deci este în FN2.
- FN3: nu există dependențe între atrbutele ne-cheie, aşadar este în FN3.

CLASA (#id\_clasa, id\_sala, id\_trainer, nume\_clasa, capacitate\_maxima) :

- FN1: toate atrbutele sunt atomice, este în FN1.
- FN2: cheia este id\_clasa, toate celelalte atrbute depind direct de ea, deci este în FN2.
- FN3: nu există dependențe între atrbutele ne-cheie, aşadar este în FN3.

REZERVARI (#id\_rezervare, id\_membru, id\_clasa, data\_rezervare, status) :

- FN1: toate atrbutele sunt atomice, este în FN1.
- FN2: cheia este id\_rezervare, toate celelalte atrbute depind direct de ea, deci este în FN2.
- FN3: nu există dependențe între atrbutele ne-cheie, aşadar este în FN3.

ECHIPAMENT (#id\_echipament, id\_sala, denumire\_echipament, tip\_echipament, status\_echipament, nr\_total\_echipament) :

- FN1: toate atrbutele sunt atomice, este în FN1.
- FN2: cheia este id\_echipament, toate celelalte atrbute depind direct de ea, deci este în FN2.
- FN3: nu există dependențe între atrbutele ne-cheie, aşadar este în FN3.

MENTENANTA\_ECHIPAMENT (#id\_echipament, data\_verificare, stare\_verificare, comentarii) :

- FN1: toate atrbutele sunt atomice, este în FN1.
- FN2: cheia este id\_echipament, toate celelalte atrbute depind direct de ea, deci este în FN2.
- FN3: am doar id\_echipament ca și cheie primară, însă același echipament poate avea mai multe mentenețe în timp. Așadar fac cheia primară compusă (id\_echipament și data\_verificare).

Rezolvare:

```
MENTENANTA_ECHIPAMENT (
    id_echipament INT,
    data_verificare DATE DEFAULT SYSDATE,
    stare_verificare VARCHAR2(100) NOT NULL,
    comentarii VARCHAR2(500),
```

```
PRIMARY KEY (id_echipament, data_verificare),
FOREIGN KEY (id_echipament) REFERENCES ECHIPAMENT(id_echipament)
);
```

## **10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).**

Codul aferent rezolvării cerinței se află și în documentul 144\_Dragunoi\_Miruna-create\_insert.txt.

```
-- CREAREA TABELELOR

--- CREATE SALA_FITNESS(#id_sala, locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)

CREATE TABLE SALA_FITNESS (
    id_sala NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    locatie VARCHAR2(50) NOT NULL,
    lant_firma VARCHAR2(50) NOT NULL,
    nr_abonati INT,
    nr_traineri INT,
    nr_echipamente INT,
    nr_telefon VARCHAR2(10) UNIQUE,
    ora_deschidere DATE,
    ora_inchidere DATE
);

--- CREATE TRAINER(#id_trainer, id_sala, nume, nr_telefon, email, specializare, salariu)

CREATE TABLE TRAINER (
    id_trainer NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_sala INT,
    nume VARCHAR2(50) UNIQUE NOT NULL,
    nr_telefon VARCHAR2(10) UNIQUE NOT NULL,
    email VARCHAR2(50) UNIQUE,
    specializare VARCHAR2(50) NOT NULL,
    salariu NUMBER(15, 2),
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala)
);

--- CREATE MEMBRU(#id_membru, id_sala, id_trainer, nume, email, nr_telefon, data_inscriere)

CREATE TABLE MEMBRU (
    id_membru NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_sala INT,
    id_trainer INT,
    nume VARCHAR2(50) UNIQUE NOT NULL,
    email VARCHAR2(50) UNIQUE,
    nr_telefon VARCHAR2(10) UNIQUE NOT NULL,
    data_inscriere DATE DEFAULT SYSDATE,
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala),
    FOREIGN KEY (id_trainer) REFERENCES TRAINER(id_trainer)
);
```

```
-- CREATE ECHIPAMENT(#id_echipament, id_sala, denumire, tip_echipament, status, nr_total)

CREATE TABLE ECHIPAMENT (
    id_echipament INT PRIMARY KEY,
    id_sala INT,
    denumire VARCHAR2(50) NOT NULL,
    tip_echipament VARCHAR2(50),
    status VARCHAR2(50),
    nr_total INT,
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala)
);

CREATE SEQUENCE seq_id_echipament
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```

--- CREATE MENTENANTA_ECHIPAMENT(#id_echipament, #data_verificare, stare_verificare, comentarii)

CREATE TABLE MENTENANTA_ECHIPAMENT (
    id_echipament INT,
    data_verificare DATE DEFAULT SYSDATE,
    stare_verificare VARCHAR2(100) NOT NULL,
    comentarii VARCHAR2(500),
    PRIMARY KEY (id_echipament, data_verificare),
    CONSTRAINT fk_mentenanta_echipament FOREIGN KEY (id_echipament)
        REFERENCES ECHIPAMENT(id_echipament)
);

--- CREATE SPECIALIZARE(#functie, incadrare, salariu)

CREATE TABLE SPECIALIZARE (
    functie VARCHAR2(50) PRIMARY KEY,
    incadrare VARCHAR2(50),
    salariu NUMBER(10, 2) NOT NULL
);

--- CREATE ANGAJAT(#id_angajat, id_sala, nume, functie, telefon, email)

CREATE TABLE ANGAJAT (
    id_angajat NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_sala INT,
    nume VARCHAR2(50) NOT NULL,
    functie VARCHAR2(50) NOT NULL,
    telefon VARCHAR2(10) UNIQUE NOT NULL,
    email VARCHAR2(50) UNIQUE,
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala),
    FOREIGN KEY (functie) REFERENCES SPECIALIZARE(functie)
);

--- CREATE TIP_ABONAMENT(#tip_abonament)

CREATE TABLE TIP_ABONAMENT (
    tip_abonament VARCHAR2(50) PRIMARY KEY
);

--- CREATE TIP_ABONAMENT_SALA(#tip_abonament, #id_sala, pret)

CREATE TABLE TIP_ABONAMENT_SALA (
    tip_abonament VARCHAR2(50),
    id_sala INT,
    pret NUMBER(10, 2) NOT NULL,
    PRIMARY KEY (tip_abonament, id_sala),
    FOREIGN KEY (tip_abonament) REFERENCES TIP_ABONAMENT(tip_abonament),
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala)
);

```

```

--- CREATE ABONAMENT(#id_abonament, id_sala, id_membru, tip_abonament, activ, data_start, data_end)

CREATE TABLE ABONAMENT (
    id_abonament NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_sala INT,
    id_membru INT,
    tip_abonament VARCHAR2(50) NOT NULL,
    activ NUMBER(1),
    data_start DATE NOT NULL,
    data_end DATE,
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala),
    FOREIGN KEY (id_membru) REFERENCES MEMBRU(id_membru),
    FOREIGN KEY (tip_abonament, id_sala) REFERENCES TIP_ABONAMENT_SALA(tip_abonament, id_sala)
);

--- CREATE PLATI_ABONAMENTE(#id_abonament, id_membru, suma_plata, data_plata, metoda_plata, status_plata)

CREATE TABLE PLATI_ABONAMENTE (
    id_abonament INT PRIMARY KEY,
    id_membru INT,
    suma_plata NUMBER(10, 2),
    data_plata DATE DEFAULT SYSDATE,
    metoda_plata VARCHAR2(50),
    status_plata VARCHAR2(50),
    FOREIGN KEY (id_abonament) REFERENCES ABONAMENT(id_abonament),
    FOREIGN KEY (id_membru) REFERENCES MEMBRU(id_membru)
);

--- CREATE CLASA(#id_clasa, id_sala, id_trainer, nume, capacitate_maxima)

CREATE TABLE CLASA (
    id_clasa NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_sala INT,
    id_trainer INT,
    nume VARCHAR2(100) NOT NULL,
    capacitate_maxima INT,
    FOREIGN KEY (id_sala) REFERENCES SALA_FITNESS(id_sala),
    FOREIGN KEY (id_trainer) REFERENCES TRAINER(id_trainer)
);

--- CREATE REZERVARE(#id_rezervare, id_membru, data_rezervare, status)

CREATE TABLE REZERVARE (
    id_rezervare NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,
    id_membru INT,
    id_clasa INT,
    data_rezervare DATE NOT NULL,
    status VARCHAR2(50),
    FOREIGN KEY (id_membru) REFERENCES MEMBRU(id_membru),
    FOREIGN KEY (id_clasa) REFERENCES CLASA(id_clasa)
);

```

```

--- TABELE ASOCIAATIVE

--- CREATE MEMBRU_CLASA(#id_membru, #id_clasa, data_inscriere, status_participare, feedback_membru)

CREATE TABLE MEMBRU_CLASA (
    id_membru INT,
    id_clasa INT,
    data_inscriere DATE DEFAULT SYSDATE,
    status_participare VARCHAR2(50),
    feedback_membru NUMBER(5),

    PRIMARY KEY (id_membru, id_clasa),
    FOREIGN KEY (id_membru) REFERENCES MEMBRU(id_membru),
    FOREIGN KEY (id_clasa) REFERENCES CLASA(id_clasa)
);

--- CREATE MEMBRU_ECHIPAMENT(#id_membru, #id_echipament, timp_utilizare, data_ultima_actualizare, intensitate, feedback_utilizare)

CREATE TABLE MEMBRU_ECHIPAMENT (
    id_membru INT,
    id_echipament INT,
    timp_utilizare INT,
    data_ultima_actualizare DATE DEFAULT SYSDATE,
    intensitate VARCHAR2(50),
    feedback_utilizare VARCHAR2(500),

    PRIMARY KEY (id_membru, id_echipament),
    FOREIGN KEY (id_membru) REFERENCES MEMBRU(id_membru),
    FOREIGN KEY (id_echipament) REFERENCES ECHIPAMENT(id_echipament)
);

--- CREATE CLASA_REZERVARE(#id_clasa, #id_rezervare, ora_rezervare, confirmare_trainer, observatii)

CREATE TABLE CLASA_REZERVARE (
    id_clasa INT,
    id_rezervare INT,
    ora_rezervare DATE,
    confirmare_trainer NUMBER(1),
    observatii VARCHAR2(500),

    PRIMARY KEY (id_clasa, id_rezervare),
    FOREIGN KEY (id_clasa) REFERENCES CLASA(id_clasa),
    FOREIGN KEY (id_rezervare) REFERENCES REZERVARE(id_rezervare)
);

```

**11. Crearea tabelelor SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabele asociative; maxim 30 de înregistrări în fiecare tabel).**

Codul aferent rezolvării cerinței se află și în documentul 144\_Dragunoi\_Miruna-create\_insert.txt.

```

--- INSERT!
--- INSERT INTO SALA_FITNESS

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Bucuresti', '18GYM', 356, 34, 457, '0345682934',
       TO_DATE('07:00', 'HH24:MI'), TO_DATE('23:59', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Cluj-Napoca', 'StrongGym', 237, 20, 300, '0756789034',
       TO_DATE('10:00', 'HH24:MI'), TO_DATE('20:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Iasi', '18GYM', 450, 56, 670, '0724536719',
       TO_DATE('08:00', 'HH24:MI'), TO_DATE('22:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Bucuresti', 'WorldClass', 467, 45, 346, '0745679874',
       TO_DATE('09:00', 'HH24:MI'), TO_DATE('21:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Piatra Neamt', 'StayFit Gym', 245, 43, 523, '0762319076',
       TO_DATE('09:30', 'HH24:MI'), TO_DATE('22:30', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Brasov', 'Matrix Gym', 450, 34, 430, '0745678912',
       TO_DATE('12:00', 'HH24:MI'), TO_DATE('00:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Onadea', 'WorldClass', 195, 19, 120, '0719872368',
       TO_DATE('11:00', 'HH24:MI'), TO_DATE('20:30', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Cluj-Napoca', 'SmartGym', 250, 21, 100, '0723456789',
       TO_DATE('06:30', 'HH24:MI'), TO_DATE('22:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Timisoara', 'StayFit', 180, 15, 90, '0744123456',
       TO_DATE('07:00', 'HH24:MI'), TO_DATE('21:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Iasi', 'WorldClass', 320, 25, 150, '0765432198',
       TO_DATE('08:00', 'HH24:MI'), TO_DATE('23:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Brasov', 'SmartGym', 160, 12, 80, '0700112233',
       TO_DATE('05:45', 'HH24:MI'), TO_DATE('20:00', 'HH24:MI'));

INSERT INTO SALA_FITNESS (locatie, lant_firma, nr_abonati, nr_traineri, nr_echipamente, nr_telefon, ora_deschidere, ora_inchidere)
VALUES ('Bucuresti', 'StayFit', 500, 30, 200, '0755987456',
       TO_DATE('06:00', 'HH24:MI'), TO_DATE('00:00', 'HH24:MI'));

COMMIT;

```

```

--- INSERT INTO TRAINER

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (2, 'Popescu Andrei', '0723456712', 'andrei.popescu@gmail.com', 'Antrenor Fitness', 4500);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (1, 'Antonescu Maria', '0756892123', 'antonescu_maria@gmail.com', 'Antrenor Sală de forță', 5000);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (3, 'Drăgan Marian', '0756781267', 'dragan.marian12@gmail.com', 'Antrenor Clasă Zumba', 4000);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (3, 'Ionescu Laura', '0746781345', 'laura.ionescu34@gmail.com', 'Antrenor Personal', 5500);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (1, 'Popa Mihai', '0723456792', 'mihai.popa98@gmail.com', 'Fitness Funcțional', 4000);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (2, 'Georgescu Andreea', '0705692310', 'andreea.georgescu@hotmail.com', 'Cardio și HIIT', 3700);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (5, 'Dumitru Radu', '0734567890', 'radu.dumitru@gmail.com', 'Bodybuilding', 4200);

INSERT INTO TRAINER (id_sala, nume, nr_telefon, email, specializare, salariu)
VALUES (4, 'Marin Alina', '0751234567', 'alina.marin@yahoo.com', 'Pilates și Stretching', 3600);

```

```

--- INSERT INTO MEMBRU

INSERT INTO MEMBRU (id_sala, id_trainer, nume, email, nr_telefon, data_inscriere)
VALUES (1, 1, 'Alexandru Popescu', 'alexandru.popescu@gmail.com', '0721345678', TO_DATE('2025-03-15', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (ID_SALA, ID_TRAINER, NUME, EMAIL, NR_TELEFON, DATA_INSCRIERE)
VALUES (1, 11, 'Elena Ionescu', 'elena.ionescu@gmail.com', '0721345679', TO_DATE('2025-04-01', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (ID_SALA, ID_TRAINER, NUME, EMAIL, NR_TELEFON, DATA_INSCRIERE)
VALUES (2, 1, 'Mihai Stan', 'mihai.stan@yahoo.com', '0721345680', TO_DATE('2025-03-18', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (ID_SALA, ID_TRAINER, NUME, EMAIL, NR_TELEFON, DATA_INSCRIERE)
VALUES (2, 4, 'Ioana Dobre', 'ioana.dobre@yahoo.com', '0721345681', TO_DATE('2025-04-04', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (id_sala, id_trainer, nume, email, nr_telefon, data_inscriere)
VALUES (3, 4, 'Vlad Marinescu', 'vlad.marinescu@gmail.com', '0721345682', TO_DATE('2025-03-20', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (ID_SALA, ID_TRAINER, NUME, EMAIL, NR_TELEFON, DATA_INSCRIERE)
VALUES (3, 15, 'Alina Florescu', 'alina.florescu@gmail.com', '0721345683', TO_DATE('2025-03-25', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (ID_SALA, ID_TRAINER, NUME, EMAIL, NR_TELEFON, DATA_INSCRIERE)
VALUES (4, 16, 'Cătălin Radu', 'catalin.radu@gmail.com', '0721345684', TO_DATE('2025-04-02', 'YYYY-MM-DD'));

INSERT INTO MEMBRU (ID_SALA, ID_TRAINER, NUME, EMAIL, NR_TELEFON, DATA_INSCRIERE)
VALUES (4, 8, 'Larisa Vasile', 'larisa.vasile@gmail.com', '0721345685', TO_DATE('2025-03-29', 'YYYY-MM-DD'));

```

```

--- INSERT INTO ECHIPAMENT

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (1, 1, 'Aparat Abdomene', 'Fortă', 'functional', 2);

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (2, 2, 'Bară de tractiune', 'Fortă', 'functional', 3);

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (3, 2, 'Bancă pentru piept', 'Fortă', 'functional', 2);

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (3, 3, 'Bicicletă eliptică', 'Cardio', 'în menenanță', 1);

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (4, 3, 'Treadmill', 'Cardio', 'functional', 4);

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (5, 4, 'Stepper', 'Cardio', 'functional', 2);

-- INSERT INTO ECHIPAMENT (ID_ECHIPAMENT, ID_SALA, DENUMIRE, TIP_ECHIPAMENT, STATUS, NR_TOTAL)
-- VALUES (6, 4, 'Ramat la cablu', 'Fortă', 'nefunctional', 1);

```

```

--- INSERT INTO MENTENANTA_ECHIPAMENT

-- INSERT INTO MENTENANTA_ECHIPAMENT (ID_ECHIPAMENT, DATA_VERIFICARE, STARE_VERIFICARE, COMENTARII)
-- VALUES (1, TO_DATE('2025-04-01', 'YYYY-MM-DD'), 'bună', 'Verificare de rutină.');

-- INSERT INTO MENTENANTA_ECHIPAMENT (ID_ECHIPAMENT, DATA_VERIFICARE, STARE_VERIFICARE, COMENTARII)
-- VALUES (2, TO_DATE('2025-04-02', 'YYYY-MM-DD'), 'bună', 'Funcționează normal.');

-- INSERT INTO MENTENANTA_ECHIPAMENT (ID_ECHIPAMENT, DATA_VERIFICARE, STARE_VERIFICARE, COMENTARII)
-- VALUES (3, TO_DATE('2025-04-02', 'YYYY-MM-DD'), 'bună', 'Fără probleme la utilizare.');

-- INSERT INTO MENTENANTA_ECHIPAMENT (ID_ECHIPAMENT, DATA_VERIFICARE, STARE_VERIFICARE, COMENTARII)
-- VALUES (4, TO_DATE('2025-04-05', 'YYYY-MM-DD'), 'defective minoră', 'Sistem de rezistență necesită înlocuire.');

-- INSERT INTO MENTENANTA_ECHIPAMENT (ID_ECHIPAMENT, DATA_VERIFICARE, STARE_VERIFICARE, COMENTARII)
-- VALUES (5, TO_DATE('2025-04-03', 'YYYY-MM-DD'), 'bună', 'Revizie efectuată.');

```

```

--- INSERT INTO SPECIALIZARE

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Receptioner', 'full-time', 4000.00);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Contabil', 'full-time', 4500);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Administrator', 'full-time', 5200);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Asistent manager', 'full-time', 4000.00);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Tehnician Echipamente', 'full-time', 3700.00);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Curătenie', 'full-time', 4600.00);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('IT Support', 'full-time', 4300);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Receptioner PT', 'part-time', 3000.00);

INSERT INTO SPECIALIZARE (functie, incadrare, salariu) VALUES ('Contabil PT', 'part-time', 4500);

```

```

--- INSERT INTO ANGAJAT

INSERT INTO ANGAJAT (id_sala, nume, functie, telefon, email)
VALUES ('ID_SALA_1', 'Andriciuc Alex', 'Receptioner', '0765235678', 'alex.andriciuc12@gmail.com');

INSERT INTO ANGAJAT (id_sala, nume, functie, telefon, email)
VALUES ('ID_SALA_2', 'Popescu Ion', 'Contabil', '0723456123', 'ion.popescu@stronggym.ro');

INSERT INTO ANGAJAT (id_sala, nume, functie, telefon, email)
VALUES ('ID_SALA_3', 'Marin Andreea', 'Administrator', '0756123498', 'andreea.marin@18gym.ro');

INSERT INTO ANGAJAT (id_sala, nume, functie, telefon, email)
VALUES ('ID_SALA_4', 'Ionescu Vlad', 'Receptioner PT', '0789456721', 'vlad.ionescu@worldclass.ro');

INSERT INTO ANGAJAT (id_sala, nume, functie, telefon, email)
VALUES ('ID_SALA_5', 'Dumitrescu Bianca', 'Asistent manager', '0712349876', 'bianca.dumitrescu@stayfit.ro');

INSERT INTO ANGAJAT (id_sala, nume, functie, telefon, email)
VALUES ('ID_SALA_6', 'Stoica Daniel', 'Tehnician Echipamente', '0745981230', 'daniel.stoica@matrixgym.ro');

```

```

--- INSERT INTO TIP_ABONAMENT

INSERT INTO TIP_ABONAMENT (TIP_ABONAMENT) VALUES ('Zi');

INSERT INTO TIP_ABONAMENT (TIP_ABONAMENT) VALUES ('Lunar');

INSERT INTO TIP_ABONAMENT (TIP_ABONAMENT) VALUES ('Anual');

INSERT INTO TIP_ABONAMENT (TIP_ABONAMENT) VALUES ('10 Sedinte');

COMMIT;

```

```

--- INSERT INTO TIP_ABONAMENT_SALA

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('Lunar', 'ID_SALA_1', 100);

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('Anual', 'ID_SALA_1', 1500);

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('Zi', 'ID_SALA_2', 90);

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('Lunar', 'ID_SALA_2', 130);

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('Anual', 'ID_SALA_2', 1999.99);

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('10 Sedinte', 'ID_SALA_3', 200);

INSERT INTO TIP_ABONAMENT_SALA (TIP_ABONAMENT, ID_SALA, PRET) VALUES ('Lunar', 'ID_SALA_3', 350);

```

```

--- INSERT INTO ABONAMENT

INSERT INTO ABONAMENT (ID_SALA, ID_MEMBRU, TIP_ABONAMENT, ACTIV, DATA_START, DATA_END)
VALUES ('ID_SALA_1', 'ID_MEMBRU_1', 'Lunar',
CASE
    WHEN SYSDATE BETWEEN TO_DATE('2025-05-01', 'YYYY-MM-DD') AND ADD_MONTHS(TO_DATE('2025-05-01', 'YYYY-MM-DD'), 1)
    THEN 1 ELSE 0
END,
DATA_START TO_DATE('2025-05-01', 'YYYY-MM-DD'), DATA_END ADD_MONTHS(TO_DATE('2025-05-01', 'YYYY-MM-DD'), 1));

INSERT INTO ABONAMENT (ID_SALA, ID_MEMBRU, TIP_ABONAMENT, ACTIV, DATA_START, DATA_END)
VALUES ('ID_SALA_1', 'ID_MEMBRU_2', 'Anual',
CASE
    WHEN SYSDATE BETWEEN TO_DATE('2025-02-01', 'YYYY-MM-DD') AND ADD_MONTHS(TO_DATE('2025-02-01', 'YYYY-MM-DD'), 1*12)
    THEN 1 ELSE 0
END,
DATA_START TO_DATE('2025-02-01', 'YYYY-MM-DD'), DATA_END ADD_MONTHS(TO_DATE('2025-02-01', 'YYYY-MM-DD'), 1*12));

INSERT INTO ABONAMENT (ID_SALA, ID_MEMBRU, TIP_ABONAMENT, ACTIV, DATA_START, DATA_END)
VALUES ('ID_SALA_2', 'ID_MEMBRU_3', 'Zi',
CASE
    WHEN SYSDATE BETWEEN TO_DATE('2025-01-23', 'YYYY-MM-DD') AND TO_DATE('2025-01-23', 'YYYY-MM-DD') + 1
    THEN 1 ELSE 0
END,
DATA_START TO_DATE('2025-01-23', 'YYYY-MM-DD'), DATA_END TO_DATE('2025-01-23', 'YYYY-MM-DD') + 1);

```

```

--- INSERT INTO PLATI_ABONAMENTE

INSERT INTO PLATI_ABONAMENTE (id_abonament, id_membru, suma_plata, data_plata, metoda_plata, status_plata)
VALUES (ID_ABONAMENT 1, ID_MEMBRU 1,
       (SELECT SUMA_PLATA PRET
        FROM TIP_ABONAMENT_SALA t
        JOIN ABONAMENT a 1<->1..n ON a.tip_abonament = t.tip_abonament AND a.id_sala = t.id_sala
        WHERE a.id_abonament = 1),
       DATA_PLATA SYSDATE,
       METODA_PLATA 'CARD',
       STATUS_PLATA 'Plată efectuată'
      );

INSERT INTO PLATI_ABONAMENTE (id_abonament, id_membru, suma_plata, data_plata, metoda_plata, status_plata)
VALUES (ID_ABONAMENT 2, ID_MEMBRU 2,
       (SELECT SUMA_PLATA PRET
        FROM TIP_ABONAMENT_SALA t
        JOIN ABONAMENT a 1<->1..n ON a.tip_abonament = t.tip_abonament AND a.id_sala = t.id_sala
        WHERE a.id_abonament = 2),
       DATA_PLATA SYSDATE,
       METODA_PLATA 'CASH',
       STATUS_PLATA 'Plată efectuată'
      );

```

```

--- INSERT INTO CLASA

INSERT INTO CLASA (id_sala, id_trainer, nume, capacitate_maxima)
VALUES (ID_SALA 5, ID_TRAINER 17, NUME 'Clase de Pilates', CAPACITATE_MAXIMA 15);

INSERT INTO CLASA (id_sala, id_trainer, nume, capacitate_maxima)
VALUES (ID_SALA 2, ID_TRAINER 6, NUME 'Spinning Intensiv', CAPACITATE_MAXIMA 20);

INSERT INTO CLASA (id_sala, id_trainer, nume, capacitate_maxima)
VALUES (ID_SALA 4, ID_TRAINER 8, NUME 'Yoga pentru începători', CAPACITATE_MAXIMA 18);

INSERT INTO CLASA (id_sala, id_trainer, nume, capacitate_maxima)
VALUES (ID_SALA 4, ID_TRAINER 8, NUME 'HIIT Avansat', CAPACITATE_MAXIMA 12);

INSERT INTO CLASA (id_sala, id_trainer, nume, capacitate_maxima)
VALUES (ID_SALA 3, ID_TRAINER 15, NUME 'Antrenament Funcțional', CAPACITATE_MAXIMA 16);

INSERT INTO CLASA (id_sala, id_trainer, nume, capacitate_maxima)
VALUES (ID_SALA 3, ID_TRAINER 3, NUME 'Zumba Fitness', CAPACITATE_MAXIMA 22);

```

```

--- INSERT INTO REZERVARE

▼ INSERT INTO REZERVARE (id_membru, id_clasa, data_rezervare, status)
VALUES (ID_MEMBRU 1, ID_CLASA 6, DATA_REZERVARE SYSDATE, STATUS 'Participare confirmată');

▼ INSERT INTO REZERVARE (id_membru, id_clasa, data_rezervare, status)
VALUES (ID_MEMBRU 4, ID_CLASA 9, DATA_REZERVARE SYSDATE, STATUS 'Participare confirmată');

▼ INSERT INTO REZERVARE (id_membru, id_clasa, data_rezervare, status)
VALUES (ID_MEMBRU 9, ID_CLASA 11, DATA_REZERVARE SYSDATE, STATUS 'Participare în așteptare');

▼ INSERT INTO REZERVARE (id_membru, id_clasa, data_rezervare, status)
VALUES (ID_MEMBRU 15, ID_CLASA 6, DATA_REZERVARE SYSDATE, STATUS 'Participare confirmată');

▼ INSERT INTO REZERVARE (id_membru, id_clasa, data_rezervare, status)
VALUES (ID_MEMBRU 7, ID_CLASA 17, DATA_REZERVARE SYSDATE, STATUS 'Participare confirmată');

▼ INSERT INTO REZERVARE (id_membru, id_clasa, data_rezervare, status)
VALUES (ID_MEMBRU 11, ID_CLASA 13, DATA_REZERVARE SYSDATE, STATUS 'Participare în așteptare');

```

```

--- INSERT INTO MEMBRU_CLASA

INSERT INTO MEMBRU_CLASA (id_membru, id_clasa, data_inscriere, status_participare, feedback_membru)
VALUES (ID_MEMBRU_1, ID_CLASA_6, DATA_INSCRIERE SYSDATE, STATUS_PARTICIPARE 'Activ', FEEDBACK_MEMBRU_5);

INSERT INTO MEMBRU_CLASA (id_membru, id_clasa, data_inscriere, status_participare, feedback_membru)
VALUES (ID_MEMBRU_4, ID_CLASA_9, DATA_INSCRIERE SYSDATE, STATUS_PARTICIPARE 'Activ', FEEDBACK_MEMBRU_5);

INSERT INTO MEMBRU_CLASA (id_membru, id_clasa, data_inscriere, status_participare, feedback_membru)
VALUES (ID_MEMBRU_9, ID_CLASA_11, DATA_INSCRIERE SYSDATE, STATUS_PARTICIPARE 'Va urma', FEEDBACK_MEMBRU_4);

INSERT INTO MEMBRU_CLASA (id_membru, id_clasa, data_inscriere, status_participare, feedback_membru)
VALUES (ID_MEMBRU_15, ID_CLASA_6, DATA_INSCRIERE SYSDATE, STATUS_PARTICIPARE 'Activ', FEEDBACK_MEMBRU_5);

INSERT INTO MEMBRU_CLASA (id_membru, id_clasa, data_inscriere, status_participare, feedback_membru)
VALUES (ID_MEMBRU_7, ID_CLASA_17, DATA_INSCRIERE SYSDATE, STATUS_PARTICIPARE 'Va urma', FEEDBACK_MEMBRU_3);

```

```

--- INSERT INTO MEMBRU_ECHIPAMENT

INSERT INTO MEMBRU_ECHIPAMENT (id_membru, id_echipament, timp_utilizare, data_ultima_actualizare, intensitate, feedback_utilizare)
VALUES (ID_MEMBRU_3, ID_ECHIPAMENT_2, TEMP_UTILIZARE_15, DATA_ULTIMA_ACTUALIZARE SYSDATE, INTENSITATE 'Intensiv', FEEDBACK_UTILIZARE 'Echipament bun');

INSERT INTO MEMBRU_ECHIPAMENT (id_membru, id_echipament, timp_utilizare, data_ultima_actualizare, intensitate, feedback_utilizare)
VALUES (ID_MEMBRU_6, ID_ECHIPAMENT_7, TEMP_UTILIZARE_30, DATA_ULTIMA_ACTUALIZARE SYSDATE, INTENSITATE 'Moderat', FEEDBACK_UTILIZARE 'Functioneaza bine, dar usor uzat');

INSERT INTO MEMBRU_ECHIPAMENT (id_membru, id_echipament, timp_utilizare, data_ultima_actualizare, intensitate, feedback_utilizare)
VALUES (ID_MEMBRU_10, ID_ECHIPAMENT_8, TEMP_UTILIZARE_20, DATA_ULTIMA_ACTUALIZARE SYSDATE, INTENSITATE 'Redusă', FEEDBACK_UTILIZARE 'Perfect pentru incepatori');

INSERT INTO MEMBRU_ECHIPAMENT (id_membru, id_echipament, timp_utilizare, data_ultima_actualizare, intensitate, feedback_utilizare)
VALUES (ID_MEMBRU_18, ID_ECHIPAMENT_13, TEMP_UTILIZARE_15, DATA_ULTIMA_ACTUALIZARE SYSDATE, INTENSITATE 'Intensiv', FEEDBACK_UTILIZARE 'Echipament bun');

```

```

--- INSERT INTO CLASA_REZERVARE

INSERT INTO CLASA_REZERVARE (id_clasa, id_rezervare, ora_rezervare, confirmare_trainer, observatii)
VALUES (ID_CLASA_6, ID_REZERVARE_1, ORA_REZERVARE TO_DATE('16:00', 'HH24:MI'), CONFIRMARE_TRAINER_1, OBSERVATII 'Doreste sa stea mai retras');

INSERT INTO CLASA_REZERVARE (id_clasa, id_rezervare, ora_rezervare, confirmare_trainer, observatii)
VALUES (ID_CLASA_9, ID_REZERVARE_2, ORA_REZERVARE TO_DATE('17:50', 'HH24:MI'), CONFIRMARE_TRAINER_1, OBSERVATII 'Doreste sa stea mai in public');

INSERT INTO CLASA_REZERVARE (id_clasa, id_rezervare, ora_rezervare, confirmare_trainer, observatii)
VALUES (ID_CLASA_11, ID_REZERVARE_3, ORA_REZERVARE TO_DATE('08:30', 'HH24:MI'), CONFIRMARE_TRAINER_1, OBSERVATII 'Va intarzia 5 minute');

INSERT INTO CLASA_REZERVARE (id_clasa, id_rezervare, ora_rezervare, confirmare_trainer, observatii)
VALUES (ID_CLASA_6, ID_REZERVARE_4, ORA_REZERVARE TO_DATE('10:00', 'HH24:MI'), CONFIRMARE_TRAINER_0, OBSERVATII 'Trainer indisponibil');

```

Imaginiile cu tabelele completate cu datele coerente:

ABONAMENT:

ID_ABONAMENT	ID_SALA	ID_MEMBRU	TIP_ABONAMENT	ACTIV	DATA_START	DATA_END
1	1	1	1 Lunar	1	2025-05-01	2025-06-01
2	2	1	2 Anual	1	2025-02-01	2026-02-01
3	3	2	3 Zi	0	2025-01-23	2025-01-24
4	4	2	4 Anual	1	2025-02-21	2026-02-21
5	5	3	5 10 Ședințe	1	2025-04-15	<null>
6	6	3	6 Lunar	0	2025-03-11	2025-04-11
7	7	4	7 Anual	1	2025-02-21	2026-02-21
8	8	4	8 Lunar	0	2025-03-01	2025-04-01
9	9	5	9 Anual	0	2024-12-12	2026-12-12
10	10	5	10 Zi	0	2025-01-19	2025-02-22
11	11	6	11 Anual	1	2025-01-15	2026-01-15
12	12	6	12 Lunar	0	2025-03-08	2025-04-08
13	13	7	13 Anual	1	2025-05-01	2026-05-01
14	14	7	14 Lunar	1	2025-04-21	2025-05-21
15	15	8	15 10 Ședințe	1	2025-02-01	<null>
16	16	8	16 Lunar	0	2025-03-13	2025-04-13
17	17	9	17 10 Ședințe	1	2025-03-01	<null>
18	18	9	18 Lunar	0	2025-01-21	2025-02-21
19	19	10	19 Lunar	0	2025-04-12	2025-05-12
20	20	10	20 10 Ședințe	1	2025-04-12	<null>
21	21	11	21 Lunar	0	2025-03-14	2025-04-14
22	22	11	22 Anual	1	2025-01-18	2026-01-18
23	23	12	23 Lunar	1	2025-04-12	<null>
24	24	12	24 Anual	1	2025-02-12	2026-02-12

## ANGAJAT:

	ID_ANGAJAT	ID_SALA	NUME	FUNCTIE	TELEFON	EMAIL
1		1	Andriciuc Alex	Receptioner	0765235678	alex.andriciuc12@gmail.com
2		2	Popescu Ion	Contabil	0723456123	ion.popescu@stronggym.ro
3		3	Marin Andreea	Administrator	0756123498	andreea.marin@18gym.ro
4		4	Ionescu Vlad	Receptioner PT	0789456721	vlad.ionescu@worldclass.ro
5		5	Dumitrescu Bianca	Asistent manager	0712349876	bianca.dumitrescu@stayfit.ro
6		6	Stoica Daniel	Tehnician Echipamente	0745981230	daniel.stoica@matrixgym.ro
7		7	Ilie Carmen	Asistent manager PT	0767812345	carmen.ilie@18gym.ro
8		8	Toma Răzvan	Curățenie PT	0728123456	razvan.toma@stronggym.ro
9		9	Cristea Eliza	Receptioner PT	0734123890	eliza.cristea@18gym.ro
10		10	Neagu Florin	Contabil	0700111122	florin.neagu@worldclass.ro
11		11	Costache Teodora	IT Support PT	0756234789	teodora.costache@stayfit.ro
12		12	Simion Paul	Tehnician Echipamente PT	0789561234	paul.simion@matrixgym.ro
13		13	Matei Ana	Receptioner	0723568912	ana.matei@worldclass.ro
14		14	Lupu Cristian	IT Support	0712784569	cristian.lupu@smartgym.ro
15		15	Vasilescu Daria	Curățenie	0700654321	daria.vasilescu@stayfit.ro
16		16	Stan Gabriel	Contabil	0754412387	gabriel.stan@worldclass.ro
17		17	Barbu Raluca	Receptioner	0734567812	raluca.barbu@smartgym.ro
18		18	Nistor Marius	Tehnician Echipamente	0767123456	marius.nistor@stayfit.ro
19		19	Dobre Ana	Asistent manager	0740987123	ana.dobre@stayfit.ro
20		20	Zamfir Oana	Administrator PT	0723345567	oana.zamfir@matrixgym.ro

## CLASA:

	ID_CLASA	ID_SALA	ID_TRAINER	NUME	CAPACITATE_MAXIMA
1		1	5	Clase de Pilates	15
2		2	2	Spinning Intensiv	20
3		3	4	Yoga pentru Începători	18
4		4	4	HIIT Avansat	12
5		5	3	Antrenament Funcțional	16
6		6	3	Zumba Fitness	22
7		7	4	Stretching & Mobilitate	14
8		8	1	TRX Suspension	10
9		9	5	Core & Abs	17
10		10	10	Box Fitness	20
11		11	6	Functional Cardio	15
12		12	5	BodyPump	18
13		13	7	Aerobic Dance	20
14		14	10	Cardio & Tonifiere	16
15		15	6	Circuit Training	12
16		16	12	Fitness General	15
17		17	10	Nutriție	30
18		18	8	Kinetoterapie	15
19		19	6	CrossFit	12
20		20	6	Circuit Training	12
21		21	11	Pilates Moderat	13

## CLASA\_REZERVARE:

	ID_CLASA	ID_REZERVARE	ORA_REZERVARE	CONFIRMARE_TRAINER	OBSERVATII
1		6	1 2025-05-01 16:00:00		1 Dorește să stea mai retras
2		9	2 2025-05-01 17:50:00		1 Dorește să stea mai în public
3		11	3 2025-05-01 08:30:00		1 Va întârziă 5 minute
4		6	4 2025-05-01 10:00:00		0 Trainer indisponibil
5		17	5 2025-05-01 14:15:00		1 Membru nou în clasă
6		13	6 2025-05-01 12:00:00		1 Preferă zona din față
7		17	7 2025-05-01 09:45:00		0 Trainer nu a confirmat încă
8		17	8 2025-05-01 17:30:00		1 Antrenament personalizat
9		18	9 2025-05-01 19:00:00		1 Are nevoie de stretching suplimentar
10		14	10 2025-05-01 07:00:00		1 Se antrenează pentru maraton
11		19	11 2025-05-01 15:30:00		0 Trainer a anulat participarea

## ECHIPAMENT:

	ID_ECHIPAMENT	ID_SALA	DENUMIRE	TIP_ECHIPAMENT	STATUS	NR_TOTAL
1		1	1 Aparat Abdomene	Fortă	functional	2
2		2	2 Bară de tractiune	Fortă	functional	3
3		3	2 Bancă pentru piept	Fortă	functional	2
4		4	3 Bicicletă eliptică	Cardio	în menenanță	1
5		5	3 Treadmill	Cardio	functional	4
6		6	4 Stepper	Cardio	functional	2
7		7	4 Ramat la cablu	Fortă	nefunctional	1
8		8	5 Gantere	Fortă	functional	10
9		9	6 Kettlebell	Fortă	functional	8
10		10	7 Bicicletă spinning	Cardio	functional	3
11		11	7 Crosstrainer	Cardio	în menenanță	1
12		12	8 Aparat picioare	Fortă	functional	2
13		13	9 Aparat abdomen	Fortă	functional	1
14		14	10 Cordon de sărit	Cardio	functional	5
15		15	12 Masă extensiile picioare	Fortă	functional	2
16		16	12 Bancă Scott	Fortă	nefunctional	1
17		17	11 Banda alergare	Cardio	functional	4
18		18	11 Aparat triceps	Fortă	functional	1
19		19	6 Sac de box	Fortă	functional	2
20		20	8 Cadran multifuncțional	Fortă	în menenanță	1
21		21	8 Bicicletă magnetică	Cardio	functional	2

## MEMBRU:

	ID_MEMBRU	ID_SALA	ID_TRAINER	NUME	EMAIL	NR_TELEFON	DATA_INSCRIERE
1		1	1	2 Alexandru Popescu	alexandru.popescu@gmail.com	0721345678	2025-03-15
2		2	1	11 Elena Ionescu	elena.ionescu@gmail.com	0721345679	2025-04-01
3		3	2	1 Mihai Stan	mihai.stan@yahoo.com	0721345680	2025-03-18
4		4	2	6 Ioana Dobre	ioana.dobre@yahoo.com	0721345681	2025-04-04
5		5	3	4 Vlad Marinescu	vlad.marinescu@gmail.com	0721345682	2025-03-20
6		6	3	15 Alina Florescu	alina.florescu@gmail.com	0721345683	2025-03-25
7		7	4	16 Cătălin Radu	catalin.radu@gmail.com	0721345684	2025-04-02
8		8	4	8 Larisa Vasile	larisa.vasile@gmail.com	0721345685	2025-03-29
9		9	5	17 Bogdan Ilie	bogdan.ilie@gmail.com	0721345686	2025-03-12
10		10	5	7 Oana Neagu	oana.neagu@gmail.com	0721345687	2025-04-03
11		11	6	9 Cristian Nistor	cristian.nistor@gmail.com	0721345688	2025-03-22
12		12	6	18 Gabriela Stoica	gabriela.stoica@gmail.com	0721345689	2025-04-05
13		13	7	10 Robert Enache	robert.enache@gmail.com	0721345690	2025-03-27
14		14	7	<null> Irina Pavel	irina.pavel@gmail.com	0721345691	2025-04-06
15		15	8	11 Florin Călin	florin.calin@gmail.com	0721345692	2025-03-30
16		16	8	<null> Anca Petrescu	anca.petrescu@yahoo.com	0721345693	2025-04-07
17		17	9	<null> Sorin Gheorghe	sorin.gheorghe@yahoo.com	0721345694	2025-03-26
18		18	9	<null> Roxana Drăgan	roxana.dragan@gmail.com	0721345695	2025-04-08
19		19	10	19 Răzvan Ciobanu	razvan.ciobanu@gmail.com	0721345696	2025-03-21
20		20	10	12 Silvia Bălan	silvia.balan@gmail.com	0721345697	2025-04-09
21		21	11	<null> Tudor Damian	tudor.damian@yahoo.com	0721345698	2025-04-10
22		22	11	<null> Dana Anghel	dana.anghel@gmail.com	0721345699	2025-04-11
23		23	12	20 Alexandru Moldovan	alex.moldovan@yahoo.com	0721345700	2025-04-12
24		24	12	15 Bianca Răileanu	bianca.raileanu@gmail.com	0721345701	2025-04-13

## MEMBRU\_CLASA:

	ID_MEMBRU	ID_CLASA	DATA_INSCRIERE	STATUS_PARTICIPARE	FEEDBACK_MEMBRU
1		1	6 2025-05-15 20:42:08	Activ	5
2		4	9 2025-05-15 20:42:08	Activ	5
3		9	11 2025-05-15 20:42:08	Va urma	4
4		15	6 2025-05-15 20:42:08	Activ	5
5		7	17 2025-05-15 20:42:08	Va urma	3
6		11	13 2025-05-15 20:42:08	Activ	5
7		16	17 2025-05-15 20:42:08	Activ	5
8		21	17 2025-05-15 20:42:08	Activ	5
9		23	10 2025-05-15 20:42:08	Activ	5
10		16	14 2025-05-15 20:42:08	Activ	5
11		24	19 2025-05-15 20:42:08	Activ	4

## MEMBRU\_ECHIPAMENT:

	ID_MEMBRI	ID_ECHIPAMENT	TIMP_UTILIZARE	DATA_ULTIMA_ACTUALIZARE	INTENSITATE	FEEDBACK_UTILIZARE
1	3	2	15	2025-05-15 20:44:37	Intensiv	Echipament bun
2	8	7	30	2025-05-15 20:44:37	Moderată	Functionează bine, dar ușor uzat
3	10	8	20	2025-05-15 20:44:37	Redusă	Perfect pentru începători
4	18	13	15	2025-05-15 20:44:37	Intensiv	Echipament bun
5	14	11	40	2025-05-15 20:44:37	Intensiv	Ideal pentru antrenamente cardio
6	23	15	25	2025-05-15 20:44:37	Moderată	Destul de confortabil
7	19	14	35	2025-05-15 20:44:37	Intensiv	Necesită menenanță la cabluri
8	14	10	18	2025-05-15 20:44:37	Redusă	Ușor de folosit și silentios
9	16	21	50	2025-05-15 20:44:37	Intensiv	Excelent pentru muscolatură
10	19	12	10	2025-05-15 20:44:37	Redusă	Feedback slab, dar util în incălzire
11	3	3	50	2025-05-15 20:44:37	Intensiv	Excelent pentru muscolatură

## MENTENANTA\_ECHIPAMENT:

	ID_ECHIPAMENT	DATA_VERIFICARE	STARE_VERIFICARE	COMENTARII
1		1 2025-04-01	bună	Verificare de rutină.
2		2 2025-04-02	bună	Functionează normal.
3		3 2025-04-02	bună	Fără probleme la utilizare.
4		4 2025-04-05	defectiune minoră	Sistem de rezistență necesită înlocuire.
5		5 2025-04-03	bună	Revizie efectuată.
6		6 2025-04-06	bună	Curătare și lubrifiere completă.
7		7 2025-04-07	nefuncțional	Cablu rupt, necesită înlocuire.
8		8 2025-04-01	excelentă	Fără semne de uzură.
9		9 2025-04-01	bună	Verificare generală efectuată.
10		10 2025-04-04	bună	Functionează fără probleme.
11		11 2025-04-04	defectiune	Pedale blocate, în așteptarea reparației.
12		12 2025-04-03	bună	Testat complet, funcțional.
13		13 2025-04-03	bună	Curătat și verificat.
14		14 2025-04-05	excelentă	Nicio problemă raportată.
15		15 2025-04-06	bună	Functionează normal.
16		16 2025-04-06	nefuncțional	Spătar rupt, neutilizabil.
17		17 2025-04-01	bună	Verificat complet.
18		18 2025-04-01	bună	Sistemul de cabluri este stabil.
19		19 2025-04-03	bună	Fără uzură semnificativă.
20		20 2025-04-05	defectiune minoră	Piesă de susținere slăbită.
21		21 2025-04-02	bună	Revizie finalizată.

## PLATI\_ABONAMENTE:

	ID_ABONAMENT	ID_MEMBRI	SUMA_PLATA	DATA_PLATA	METODA_PLATA	STATUS_PLATA
1		1	100.00	2025-05-14 01:03:24	CARD	Plată efectuată
2		2	1500.00	2025-05-14 01:03:24	CASH	Plată efectuată
3		3	90.00	2025-05-14 01:03:24	CARD	Plată eşuată
4		4	1999.99	2025-05-14 01:03:24	CARD	Plată eşuată
5		5	200.00	2025-05-14 01:03:24	CARD	În decurs de plată
6		6	350.00	2025-05-14 01:03:24	CASH	Plată efectuată
7		7	2000.00	2025-05-14 01:03:24	CARD	Plată eşuată
8		8	300.00	2025-05-14 01:03:24	CARD	Plată efectuată
9		9	2300.00	2025-05-14 01:03:24	CASH	Plată efectuată
10		10	65.00	2025-05-14 01:03:24	CARD	Plată eşuată
11		11	1499.99	2025-05-14 01:03:24	CARD	În decurs de plată
12		12	350.00	2025-05-14 01:03:24	CASH	Plată efectuată
13		13	2500.00	2025-05-14 01:03:24	CARD	Plată efectuată
14		14	500.00	2025-05-14 01:03:24	CASH	Plată efectuată
15		15	150.00	2025-05-14 01:03:24	CARD	Plată eşuată
16		16	200.00	2025-05-14 01:03:24	CASH	Plată efectuată
17		17	200.00	2025-05-14 01:03:24	CARD	În decurs de plată
18		18	350.00	2025-05-14 01:03:24	CARD	Plată eşuată
19		19	500.00	2025-05-14 01:03:24	CARD	Plată efectuată
20		20	400.00	2025-05-14 01:03:24	CASH	Plată efectuată
21		21	499.99	2025-05-14 01:03:24	TRANSFER BANCAR	Plată efectuată
22		22	700.00	2025-05-14 01:03:24	TRANSFER BANCAR	În decurs de plată
23		23	320.00	2025-05-14 01:03:24	CARD	Plată efectuată
24		24	1200.00	2025-05-14 01:03:24	CASH	Plată efectuată

## REZERVARE:

	ID_REZERVARE	ID_MEMBRU	ID_CLASA	DATA_REZERVARE	STATUS
1		1	1	6 2025-05-14 01:07:03	Participare confirmată
2		2	4	9 2025-05-14 01:07:03	Participare confirmată
3		3	9	11 2025-05-14 01:07:03	Participare în aşteptare
4		4	15	6 2025-05-14 01:07:03	Participare confirmată
5		5	7	17 2025-05-14 01:07:03	Participare confirmată
6		6	11	13 2025-05-14 01:07:03	Participare în aşteptare
7		7	16	17 2025-05-14 01:07:03	Participare confirmată
8		8	21	17 2025-05-14 01:07:04	Participare confirmată
9		9	23	10 2025-05-14 01:07:04	Participare confirmată
10		10	16	14 2025-05-14 01:07:04	Participare confirmată
11		11	24	19 2025-05-14 01:07:04	Participare confirmată
12		21	24	11 2025-05-14 13:58:01	Participare confirmată
13		22	23	11 2025-01-15 00:35:11	Participare confirmată

## SALA\_FITNESS:

	ID_SALA	LOCATIE	LANT_FIRMA	NR_ABONATI	NR_TRAINERI	NR_EQUIPAMENTE	NR_TELEFON	ORA_DESCHIDERE	ORA_INCHIDERE
1	1	Bucuresti	18GYM	2	34	457	0345602934	2025-05-01 07:00:00	2025-05-01 23:59:00
2	2	Cluj-Napoca	StrongGym	237	28	308	0756789034	2025-05-01 10:00:00	2025-05-01 20:00:00
3	3	Iasi	18GYM	450	56	670	0724536719	2025-05-01 08:00:00	2025-05-01 22:00:00
4	4	Bucuresti	WorldClass	467	45	346	0745679874	2025-05-01 09:00:00	2025-05-01 21:00:00
5	5	Piatra Neamt	StayFit Gym	245	43	523	07623519076	2025-05-01 09:30:00	2025-05-01 22:30:00
6	6	Brasov	Motrix Gym	450	34	430	0745678912	2025-05-01 12:00:00	2025-05-01
7	7	Oradea	WorldClass	2	19	120	0719872368	2025-05-01 11:00:00	2025-05-01 20:30:00
8	8	Cluj-Napoca	SmartGym	250	21	100	07234656789	2025-05-01 06:30:00	2025-05-01 22:00:00
9	9	Timisoara	StayFit	180	15	98	0744123456	2025-05-01 07:00:00	2025-05-01 21:00:00
10	10	Iasi	WorldClass	329	25	150	0765432198	2025-05-01 08:00:00	2025-05-01 23:00:00
11	11	Brasov	SmartGym	160	12	80	0780112233	2025-05-01 05:45:00	2025-05-01 20:00:00
12	12	Bucuresti	StayFit	500	30	200	0755987456	2025-05-01 06:00:00	2025-05-01

## SPECIALIZARE:

	FUNCTIE	INCADRARE	SALARIU
1	Receptioner	full-time	4000.00
2	Contabil	full-time	4500.00
3	Administrator	full-time	5200.00
4	Asistent manager	full-time	4000.00
5	Tehnician Echipamente	full-time	3700.00
6	Curătenie	full-time	4600.00
7	IT Support	full-time	4300.00
8	Receptioner PT	part-time	3000.00
9	Contabil PT	part-time	4500.00
10	Administrator PT	part-time	4300.00
11	Asistent manager PT	part-time	3000.00
12	Tehnician Echipamente PT	part-time	3000.00
13	Curătenie PT	part-time	3600.00
14	IT Support PT	part-time	3300.00

## TIP\_ABONAMENT:

	TIP_ABONAMENT
1	10 Sedințe
2	Anual
3	Lunar
4	Zi

### TIP\_ABONAMENT\_SALA:

	TIP_ABONAMENT	ID_SALA	PRET
1	Lunar	1	110.00
2	Anual	1	1650.00
3	Zi	2	99.00
4	Lunar	2	143.00
5	Anual	2	2199.99
6	10 Şedințe	3	178.20
7	Lunar	3	311.85
8	Lunar	4	238.43
9	Anual	4	1589.50
10	Zi	5	71.50
11	Lunar	5	495.00
12	Anual	5	2530.00
13	Lunar	6	385.00
14	Anual	6	1649.99
15	Lunar	7	500.00
16	Anual	7	2500.00
17	Lunar	8	178.20
18	10 Şedințe	8	133.65
19	10 Şedințe	9	200.00
20	Lunar	9	350.00
21	Anual	9	2499.99
22	10 Şedințe	10	440.00
23	Lunar	10	550.00
24	Lunar	11	499.99
25	Anual	11	700.00
26	10 Şedințe	12	119.21
27	Lunar	12	254.32
28	Anual	12	953.70

### TRAINER:

ID_TRAINER	ID_SALA	NUME	NR_TELEFON	EMAIL	SPECIALIZARE	SALARIU
1	1	Popescu Andrei	0723456712	andrei.popescu@gmail.com	Antrenor Fitness	4500.00
2	2	Antonescu Maria	0756892123	antonescu_maria@gmail.com	Antrenor Sală de forță	3000.00
3	3	Drăgan Marian	0756781267	dragan.marian12@gmail.com	Antrenor Clasă Zumba	5520.00
4	4	Ionescu Laura	0746781345	laura.ionescu34@gmail.com	Antrenor Personal	3500.00
5	5	Popa Mihai	0723456792	mihai.popa98@gmail.com	Fitness Funcțional	4000.00
6	6	Georgescu Andreea	0765892310	andreea.georgescu@hotmail.com	Cardio și HIIT	4255.00
7	7	Dumitru Radu	0734567890	radu.dumitru@gmail.com	Bodybuilding	5796.00
8	8	Marin Alina	0751234567	alina.marin@yahoo.com	Pilates și Stretching	4140.00
9	9	Stan Tudor	0789456123	tudor.stan@18gym.ro	Antrenor CrossFit	5280.00
10	10	Iliescu Raluca	0712345678	raluca.iliescu@worldclass.ro	Antrenor Aerobic	4554.00
11	11	Petrescu Alexandru	0700112233	alex.petrescu@smartgym.ro	Kinetoterapie	3900.00
12	12	Toma Bianca	0723456888	bianca.toma@yahoo.com	Nutritie & Fitness	4715.00
13	13	Enache Vlad	0755123456	vlad.enache@gmail.com	Fitness General	3700.00
14	14	Mihăilescu Sorin	0721345678	sorin.mihalescu@gmail.com	BodyPump	3800.00
15	15	Tudose Anca	0745678213	anca.tudose@18gym.ro	Antrenor Funcțional	3900.00
16	16	Costache Liviu	0750001234	liviu.costache@gmail.com	Stretching & Mobilitate	3450.00
17	17	Rusu Adriana	0711456789	adriana.rusu@yahoo.com	Pilates	3600.00
18	18	Dobre Cristian	0733211122	cristian.dobre@crossfit.ro	CrossFit	5460.00
19	19	Lungu Elena	0788012345	elena.lungu@gmail.com	Fitness General	5175.00
20	20	Matei Cătălin	0767456789	catalin.matei@stayfit.ro	Personal Trainer	4200.00
21	21	Dragu Maria	0767456780	dragu_mari@gmail.ro	Pilates	2500.00

**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

- a) subcereri sincronizate în care intervin cel puțin 3 tabele.
- b) subcereri nesincronizate în clauza FROM.
- c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza HAVING).
- d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri).
- e) utilizarea a cel puțin 2 funcții pe siruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE.
- f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH).

**Observație:** Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.

Codul pentru cerințele de mai jos se află în fișierul 144\_Dragunoi\_Miruna-select\_update\_delete.txt.

**Cerinta 1:** Afipați membrii activi care au făcut cel puțin o rezervare confirmată în ultima lună, împreună cu numele antrenorului lor preferat și categoria de activitate. Rezultatele vor fi ordonate după numărul de rezervări.

Subpunctele: a), b), c), d), e), f)

```
WITH membri_activi AS (
    SELECT
        m.id_membru,
        m.nume,
        COUNT(r.id_rezervare) AS nr_rezervari
    FROM MEMBRU m
    JOIN REZERVARE r [1..n] ON m.id_membru = r.id_membru
    WHERE r.status = 'Participare confirmată'
        AND r.data_rezervare >= ADD_MONTHS(SYSDATE, -1)
    GROUP BY m.id_membru, m.nume
    HAVING COUNT(r.id_rezervare) >= 0
)
SELECT
    UPPER(SUBSTR(ma.nume, 1, 15)) AS nume_scurt,
    LOWER(m.email) AS email,
    ma.nr_rezervari,
    NVL(
        SELECT t.nume
        FROM (
            SELECT tr.nume, COUNT(*) cnt
            FROM REZERVARE r2
            JOIN CLASA c [1..n] ON r2.id_clasa = c.id_clasa
            JOIN TRAINER tr [1..n] ON c.id_trainer = tr.id_trainer
            WHERE r2.id_membru = ma.id_membru
            GROUP BY tr.nume
            ORDER BY cnt DESC
        ) t
        WHERE ROWNUM = 1
    ), 'Fără antrenor' AS antrenor_favorit,
    TO_CHAR(SYSDATE, 'DD-MM-YYYY') AS data_raport
FROM membri_activi ma
JOIN MEMBRU m ON ma.id_membru = m.id_membru
JOIN SALA_FITNESS sf [1..n] ON m.id_sala = sf.id_sala
```

```

WHERE
    EXISTS (
        SELECT 1
        FROM MEMBRU_CLASA mc
        JOIN CLASA c [1..n<->1] ON mc.id_clasa = c.id_clasa
        JOIN TRAINER t [1..n<->1] ON c.id_trainer = t.id_trainer
        WHERE mc.id_membru = ma.id_membru
        | AND mc.status_participare = 'Activ'
    )
    AND ma.id_membru IN (
        SELECT r.id_membru
        FROM REZERVARE r
        GROUP BY r.id_membru
        HAVING COUNT(*) > (
            SELECT AVG(cnt)
            FROM (
                SELECT COUNT(*) cnt
                FROM REZERVARE
                WHERE status = 'Participare confirmată'
                GROUP BY id_membru
            )
        )
    )
ORDER BY
    ma.nr_rezervari DESC,
    NVL(ma.nume, 'ZZZ')
FETCH FIRST 10 ROWS ONLY;

```

Rezultat:

```

33   FROM (
34     SELECT tr.num, COUNT(*) cnt
35     FROM REZERVARE r2
36     JOIN CLASA c [1..n<->1] ON r2.id_clasa = c.id_clasa
37     JOIN TRAINER tr [1..n<->1] ON c.id_trainer = tr.id_trainer
38     WHERE r2.id_membru = ma.id_membru
39     GROUP BY tr.num
40     ORDER BY cnt DESC
41   ) t
42   WHERE ROWNUM = 1
43 ), 'Fără antrenor' AS antrenor_favorit,
44   TO_CHAR(SYSDATE, 'DD-MM-YYYY') AS data_report
45 FROM membri_activi ma
46 JOIN MEMBRU m ON ma.id_membru = m.id_membru
47 JOIN SALA_FITNESS sf [1..n<->1] ON m.id_sala = sf.id_sala
48 WHERE
49   EXISTS (
50     SELECT 1
51     FROM MEMBRU_CLASA mc
52     JOIN CLASA c [1..n<->1] ON mc.id_clasa = c.id_clasa
53     JOIN TRAINER t [1..n<->1] ON c.id_trainer = t.id_trainer
54     WHERE mc.id_membru = ma.id_membru
55     | AND mc.status_participare = 'Activ'
56   )
57   AND ma.id_membru IN (
58     SELECT r.id_membru
59     FROM REZERVARE r
60     GROUP BY r.id_membru
61     HAVING COUNT(*) > (
62       SELECT AVG(cnt)
63       FROM (
64         SELECT COUNT(*) cnt
65         FROM REZERVARE
66         WHERE status = 'Participare confirmată'
67         GROUP BY id_membru
68       )
69     )
70   )
71 ORDER BY
72   ma.nr_rezervari DESC,
73   NVL(ma.num, 'ZZZ')
74 FETCH FIRST 10 ROWS ONLY;

```

	NUME_SCURT	EMAIL	NR_REZERVARI	ANTRENOR_FAVORIT	DATA_REPORT
1	ANCA PETRESCU	anca.petrescu@yahoo.com	2	Lungu Elena	15-05-2025
2	BIANCA RĂILEANU	bianca.raileanu@gmail.com	2	Dobre Cristian	15-05-2025
3	ALEXANDRU MOLDO	alex.moldovan@yahoo.com	1	Lungu Elena	15-05-2025

**Cerinta 2:** Pentru fiecare sala fitness, calculati numarul total de plati efectuate in ultimele 6 luni si suma totala platita. Afisati doar salile care au incasat peste 5000 RON. Salile vor fi sortate dupa suma incasata descrescator.

Subpunctele: f), e), c)

```

WITH plati_recente AS (
    SELECT
        p.id_membru,
        p.id_abonament,
        a.id_sala,
        p.suma_plata,
        p.data_plata
    FROM
        PLATI_ABONAMENTE p
    JOIN ABONAMENT a [1<->1] ON p.id_abonament = a.id_abonament
    WHERE
        MONTHS_BETWEEN(SYSDATE, p.data_plata) <= 6
)
SELECT
    s.id_sala,
    s.locatie,
    s.lant_firma,
    COUNT(pr.suma_plata) AS nr_plati,
    SUM(pr.suma_plata) AS total_incasari
FROM
    plati_recente pr
JOIN
    SALA_FITNESS s ON pr.id_sala = s.id_sala
GROUP BY
    s.id_sala, s.locatie, s.lant_firma
HAVING
    SUM(pr.suma_plata) > 10
ORDER BY
    total_incasari DESC;

```

Rezultat:

The screenshot shows the Oracle SQL Developer interface with the query editor and results viewer. The query is identical to the one above, designed to find salas fitness with total payments over 10k in the last 6 months, sorted by total income in descending order. The results table shows 12 rows of data.

ID_SALA	LOCATIE	LANT_FIRMA	NR_PLATI	TOTAL_INCASARI
7	Oradea	WorldClass	2	3000
5	Piatra Neamt	StayFit Gym	2	2365
4	Bucuresti	WorldClass	2	2300
2	Cluj-Napoca	StrongGym	2	2089.99
6	Brașov	Matrix Gym	2	1849.99
1	Bucuresti	18GYM	2	1600
12	Bucuresti	StayFit	2	1520
11	Brașov	SmartGym	2	1199.99
10	Iasi	WorldClass	2	900
3	Iasi	18GYM	2	550
9	Timisoara	StayFit	2	550
8	Cluj-Napoca	SmartGym	2	350

**CERINTA 3:** Afisati pentru fiecare membru activ (care are un abonament activ), numele, emailul, tipul abonamentului, valoarea totala platita, data celei mai recente plati, statusul platilor, si lungimea numelui membrului. Sa fie afisati doar membrii care au platit mai mult decat media tuturor platilor, ordonati descrescator dupa suma totala.

Subpunctele: a), b), c), d), e), f)

```

119 WITH total_plati_membri AS (
120     SELECT
121         m.id_membru,
122         m.nume AS nume_membru,
123         m.email,
124         a.tip_abonament,
125         SUM(p.suma_plata) AS total_plata,
126         MAX(p.data_plata) AS ultima_plata
127     FROM
128         MEMBRU m
129         JOIN ABONAMENT a [1<->1..n] ON m.id_membru = a.id_membru
130         JOIN PLATI_ABONAMENTE p [1<->1..n] ON a.id_abonament = p.id_abonament
131     WHERE
132         a.activ = 1
133     GROUP BY
134         m.id_membru, m.nume, m.email, a.tip_abonament
135 )
136
137 SELECT
138     INITCAP(t.nume_membru) AS nume_formatat,
139     UPPER(SUBSTR(t.email, 1, INSTR(t.email, '@') - 1)) AS user_email,
140     DECODE(t.tip_abonament,
141             'Lunar', 'Abonament Lunar',
142             'Anual', 'Abonament Anual',
143             'Zil', 'Abonament pe Zil',
144             '10 Sedinte', 'Pachet Special de 10 Sedinte',
145             'Necunoscut') AS tip_abonament,
146     t.total_plata,
147     NVL(TO_CHAR(t.ultima_plata, 'DD-MM-YYYY'), 'Fără plată') AS data_ultima_plata,
148     CASE
149         WHEN t.total_plata > 2000 THEN 'FOARTE ACTIV'
150         WHEN t.total_plata BETWEEN 1500 and 2000 THEN 'ACTIV'
151         ELSE 'INACTIV'
152     END AS status_activitate,
153     LENGTH(t.nume_membru) AS lungime_nume
154
155 FROM
156     total_plati_membri t
157 WHERE
158     t.total_plata > (SELECT AVG(suma_plata) FROM PLATI_ABONAMENTE WHERE status_plata = 'Plată efectuată')
159 ORDER BY
160     t.total_plata DESC;

```

Rezultat:

	NUME_FORMATAT	USER_EMAIL	TIP_ABONAMENT	TOTAL_PLATA	DATA_ULTIMA_PLATA	STATUS_ACTIVITATE	LUNGIME_NUME
1	Robert Enache	ROBERT.ENACHE	Abonament Anual	2500	14-05-2025	FOARTE ACTIV	13
2	Cătălin Radu	CATALIN.RADU	Abonament Anual	2000	14-05-2025	ACTIV	12
3	Ioana Dobre	IOANA.DOBRE	Abonament Anual	1999,99	14-05-2025	ACTIV	11
4	Elena Ionescu	ELENA.IONESCU	Abonament Anual	1500	14-05-2025	ACTIV	13
5	Cristian Nistor	CRISTIAN.NISTOR	Abonament Anual	1499,99	14-05-2025	INACTIV	15
6	Bianca Răileanu	BIANCA.RAILEANU	Abonament Anual	1200	14-05-2025	INACTIV	15

**CERINTA 4:** Să se realizeze o analiză a sălilor de fitness care au un procent de abonamente active de cel puțin 50%. Pentru fiecare sală se vor afișa: identificatorul sălii, locația, lanțul de care aparține, numărul total de membri, prețul minim și maxim al abonamentelor, numărul de abonamente active, procentul acestora din total. Rezultatele vor fi ordonate descrescător după numărul de membri.

Subpuncte: a), c), e)

```

SELECT
    s.id_sala,
    s.locatie,
    s.lant_firma,
    COUNT(m.id_membru) AS numar_abonati,
    MIN(tip_ab.pret) AS abonament_minim,
    MAX(tip_ab.pret) AS abonament_maxim,
    SUM(CASE WHEN a.activ = 1 THEN 1 ELSE 0 END) AS abonamente_active,
    ROUND(SUM(CASE WHEN a.activ = 1 THEN 1 ELSE 0 END) * 100 / COUNT(m.id_membru), 2) AS procent_abonamente_active
FROM
    SALA_FITNESS s
LEFT JOIN MEMBRU m [1<->0..n] ON s.id_sala = m.id_sala
LEFT JOIN ABONAMENT a [1<->0..n] ON a.id_membru = m.id_membru
LEFT JOIN TIP_ABONAMENT_SALA tip_ab ON (a.id_sala = tip_ab.id_sala AND a.tip_abonament = tip_ab.tip_abonament)
GROUP BY s.id_sala, s.locatie, s.lant_firma
HAVING SUM(CASE WHEN a.activ = 1 THEN 1 ELSE 0 END) > (
    SELECT COUNT(*) * 0.5
    FROM ABONAMENT
    WHERE id_sala = s.id_sala
)
ORDER BY numar_abonati DESC;

```

Rezultat:

```

169 ✓ SELECT
170     s.id_sala,
171     s.locatie,
172     s.lant_firma,
173     COUNT(m.id_membru) AS numar_abonati,
174     MIN(tip_ab.pret) AS abonament_minim,
175     MAX(tip_ab.pret) AS abonament_maxim,
176     SUM(CASE WHEN a.activ = 1 THEN 1 ELSE 0 END) AS abonamente_active,
177     ROUND(SUM(CASE WHEN a.activ = 1 THEN 1 ELSE 0 END) * 100 / COUNT(m.id_membru), 2) AS procent_abonamente_active
178 FROM
179     SALA_FITNESS s
180 LEFT JOIN MEMBRU m [1<->0..n] ON s.id_sala = m.id_sala
181 LEFT JOIN ABONAMENT a [1<->0..n] ON a.id_membru = m.id_membru
182 LEFT JOIN TIP_ABONAMENT_SALA tip_ab ON (a.id_sala = tip_ab.id_sala AND a.tip_abonament = tip_ab.tip_abonament)
183 GROUP BY s.id_sala, s.locatie, s.lant_firma
184 HAVING SUM(CASE WHEN a.activ = 1 THEN 1 ELSE 0 END) > (
185     SELECT COUNT(*) * 0.5
186     FROM ABONAMENT
187     WHERE id_sala = s.id_sala
188 )
189 ORDER BY numar_abonati DESC;
190
191
192 --- CERINTA 5: Să se realizeze o analiză a salariilor personalului din sălile de fitness, care să prezinte pentru
193 --- fiecare trainer: informațiile de bază (nume, specializare/funcție, locație), salariul curent,
194 --- bonusurile calculate pe baza performanței și salariul final ajustat. Analiza va include doar personalul
195

```

ID_SALA	LOCATIE	LANT_FIRMA	NUMAR_ABONATI	ABONAMENT_MINIM	ABONAMENT_MAXIM	ABONAMENTE_ACTIVE	PROCENT_ABONAMENTE_ACTIVE
1	Bucuresti	186YM	2	110	1650	2	100
2	Oradea	WorldClass	2	500	2500	2	100
3	Bucuresti	StayFit	2	254.32	953.7	2	100

**CERINTA 5:** Să se realizeze o analiză a salariilor personalului din sălile de fitness, care să prezinte pentru fiecare trainer: informațiile de bază (nume, specializare/funcție, locație), salariul curent, bonusurile calculate pe baza performanței și salariul final ajustat. Analiza va include doar personalul care are o performanță peste media companiei (traineri cu număr de membri peste medie). Rezultatele vor fi ordonate descrescător după salariul final.

Subpuncte: a), b), d), e), f)

```

WITH statistici_traineri AS (
    SELECT
        t.id_trainer,
        t.nume AS nume_trainer,
        t.specializare,
        t.salariu AS salariu_baza,
        s.locatie,
        COUNT(DISTINCT m.id_membru) AS numar_membri_antrenati,
        AVG(NVL(mc.feedback_membru, 0)) AS rating_mediul,
        MONTHS_BETWEEN(SYSDATE, MIN(m.data_inscriere)) AS vechime_max_membru
    FROM
        TRAINER t
        JOIN (SELECT id_sala, locatie FROM SALA_FITNESS) s ON t.id_sala = s.id_sala
        LEFT JOIN MEMBRU m [1->0..n] ON t.id_trainer = m.id_trainer
        LEFT JOIN CLASA c [1->0..n] ON t.id_trainer = c.id_trainer
        LEFT JOIN MEMBRU_CLASA mc [1->0..n] ON c.id_clasa = mc.id_clasa
    GROUP BY
        t.id_trainer, t.nume, t.specializare, t.salariu, s.locatie
)

```

```

SELECT
    UPPER(st.nume_trainer) AS nume,
    INITCAP(st.specializare) AS specializare,
    st.locatie,
    TRUNC(st.vechime_max_membru/12) AS anii_vechime,
    TO_CHAR(ADD_MONTHS(SYSDATE, -3), 'DD-MM-YYYY') AS data_referinta,
    NVL(st.salariu_baza, 0) AS salariu_baza,
    CASE
        WHEN st.numar_membri_antrenati > 15 THEN st.salariu_baza * 0.2
        WHEN st.numar_membri_antrenati > 5 THEN st.salariu_baza * 0.1
        ELSE 0
    END AS bonus_performanta,
    DECODE(ROUND(st.rating_mediul),
        5, st.salariu_baza * 0.25,
        4, st.salariu_baza * 0.15,
        3, st.salariu_baza * 0.05,
        0) AS bonus_evaluare,
    (SELECT COUNT(DISTINCT r.id_rezervare)
    FROM REZERVARE r
    JOIN CLASA c [1..n->1] ON r.id_clasa = c.id_clasa
    JOIN MEMBRU m [1..n->1] ON r.id_membru = m.id_membru
    WHERE c.id_trainer = st.id_trainer
    AND r.status = 'Participare confirmata') AS rezervari_finalizate,
    st.salariu_baza +
    CASE
        WHEN st.numar_membri_antrenati > 1 THEN st.salariu_baza * 0.2
        WHEN st.numar_membri_antrenati > 0 THEN st.salariu_baza * 0.1
        ELSE 0
    END +
    DECODE(ROUND(st.rating_mediul),
        5, st.salariu_baza * 0.25,
        4, st.salariu_baza * 0.15,
        3, st.salariu_baza * 0.05,
        0) AS salariu_final
FROM
    statistici_traineri st
WHERE
    st.numar_membri_antrenati > (
        SELECT AVG(numar_membri)
        FROM (
            SELECT COUNT(DISTINCT m.id_membru) AS numar_membri
            FROM TRAINER t
            LEFT JOIN MEMBRU m [1->0..n] ON t.id_trainer = m.id_trainer
            GROUP BY t.id_trainer
        )
    )
ORDER BY
    salariu_final DESC;

```

## Rezultat:

```

197
198 ✓ WITH statistici_traineri AS (
199   SELECT
200     t.id_trainer,
201     t.num AS nume_trainer,
202     t.specializare,
203     t.salariu AS salariu_baza,
204     t.locatie,
205     COUNT(DISTINCT m.id_membru) AS numar_membru_atrenanti,
206     AVG((NVL(mn.feedback_membru, 0)) / 10) AS rating_mediu,
207     MONTHS_BETWEEN(SYSDATE, MIN(m.data_inscrisie)) AS vechime_max_membru
208
209   FROM
210     TRAINER t
211     JOIN SELECT id_sala, locatie FROM SALA_FITNESS) s ON t.id_sala = s.id_sala
212     LEFT JOIN MEMBRI m ON t.id_trainer = m.id_trainer
213     LEFT JOIN CLASA c ON t.id_sala = c.id_sala AND t.id_trainer = c.id_trainer
214     LEFT JOIN MEMBRI_CLASA mc ON m.id_membru = mc.id_membru AND c.id_clasa = mc.id_clasa
215   GROUP BY
216     t.id_trainer, t.nume, t.specializare, t.salariu, t.locatie
217
218 )
219
220 SELECT
221   UPPER(st.nume_trainer) AS nume,
222   INITCAP(st.specializare) AS specializare,
223   st.locatie,
224   TRUNC(TO_CHAR(vechime_max_membru/12)) AS ani_vechime,
225   ROUND((mc.MONTHS_BETWEEN(SYSDATE, -3)) / 12, 1) AS data_referinta,
226   NVL(sal.st.salariu_baza, 0) AS salariu_baza
227   FROM

```

	Output	Subiecte: a, b, d, e), f)	CSV							
Index	Nume	Specializare	Locatie	Ani_Vechime	Data_Refinanta	Salariu_Baza	Bonus_Performanta	Bonus_Evaluare	Rezervari_Finalizate	Salariu_Final
1	LUNGU ELENA	Fitness General	Iasi	0	15-02-2025	5175	0	1293,75	2	6982,25
2	DUMITRU RADU	Bodybuilding	Piatra Neamt	0	15-02-2025	5796	0	289,8	1	6665,4
3	ILIESCU RALUCA	Antrenor Aerobic	Oradea	0	15-02-2025	4554	0	1138,5	0	6147,9
4	DOBRE CRISTIAN	Crossfit	Brasov	0	15-02-2025	5460	0	0	1	6806
5	TOMA BIANCA	Nutritie & Fitness	Iasi	0	15-02-2025	4715	0	707,25	3	5893,75
6	STAN TUDOR	Antrenor Crossfit	Brasov	0	15-02-2025	5280	0	0	2	5808
7	POPESCU ANDREI	Antrenor Fitness	Cluj-Napoca	0	15-02-2023	4500	0	0	0	4950
8	GEORGESCU ANDREEA	Cardio Si HIIT	Cluj-Napoca	0	15-02-2025	4255	0	0	0	4686,5
9	PETRESCU ALEXANDRU	Kinetoterapie	Cluj-Napoca	0	15-02-2025	3900	0	0	0	4680
10	MATEI CATALIN	Personal Trainer	Bucuresti	0	15-02-2025	4200	0	0	0	4620
11	MARIN ALINA	Pilates Si Stretching	Bucuresti	0	15-02-2025	4140	0	0	0	4554
12	TUDOSE ANCA	Antrenor Functional	Iasi	0	15-02-2025	3900	0	0	0	4290
13	ENACHE VLAD	Fitness General	Bucuresti	0	15-02-2025	3700	0	0	0	4070
14	RUSU ADRIANA	Pilates	Piatra Neamt	0	15-02-2025	3600	0	0	0	3900
15	IONESCU LAURA	Antrenor Personal	Iasi	0	15-02-2025	3500	0	0	0	3850
16	COSTACHE LIVIU	Stretching & Mobilitate	Bucuresti	0	15-02-2025	3450	0	0	0	3795
17	ANTONESCU MARIA	Antrenor Sală De Fortă	Bucuresti	0	15-02-2025	3000	0	0	0	3300

### 13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

Codul din cerințe poate fi găsit și în fișierul 144\_Dragunoi\_Miruna-select\_update\_delete.txt.

Cerinte UPDATE:

- Actualizează salariul cu 20% pentru trainerii care au primit doar feedback pozitiv (mai mare sau egal cu 4) din partea membrilor la clasele pe care le-au susținut.

```
-- 1. Actualizează salariul cu 20% pentru trainerii care au primit doar feedback pozitiv (mai mare sau egal cu 4)
-- din partea membrilor la clasele pe care le-au susținut.

UPDATE TRAINER t
SET salariu = salariu * 1.20
WHERE id_trainer IN (
    SELECT c.id_trainer
    FROM CLASA c
    JOIN MEMBRU_CLASA mc ON c.id_clasa = mc.id_clasa
    WHERE mc.feedback_membru >= 4
    GROUP BY c.id_trainer
    HAVING COUNT(*) = (
        SELECT COUNT(*)
        FROM MEMBRU_CLASA mc2
        WHERE mc2.id_clasa IN (
            SELECT id_clasa FROM CLASA WHERE id_trainer = c.id_trainer
        )
    )
);
COMMIT;

-- 2. Recalculează și actualizează numărul de abonați pentru fiecare sală de fitness, numărând doar membrii distin-
-- care au abonamente active cu data de expirare în viitor. Actualizează doar sălile care au înregistrat cel puțin
```

```
[2025-05-15 21:02:51] Connected
C##MIRUNA> alter session set current_schema = C##MIRUNA
[2025-05-15 21:02:51] completed in 1 ms
C##MIRUNA> UPDATE TRAINER t
SET salariu = salariu * 1.20
WHERE id_trainer IN (
    SELECT c.id_trainer
    FROM CLASA c
    JOIN MEMBRU_CLASA mc ON c.id_clasa = mc.id_clasa
    WHERE mc.feedback_membru >= 4
    GROUP BY c.id_trainer
    HAVING COUNT(*) = (
        SELECT COUNT(*)
        FROM MEMBRU_CLASA mc2
        WHERE mc2.id_clasa IN (
            SELECT id_clasa FROM CLASA WHERE id_trainer = c.id_trainer
        )
    )
)
[2025-05-15 21:02:52] 6 rows affected in 14 ms
C##MIRUNA> COMMIT
[2025-05-15 21:02:52] completed in 2 ms
```

2. Recalculează și actualizează numărul de abonați pentru fiecare sală de fitness, numărând doar membrii distincti care au abonamente active cu data de expirare în viitor. Actualizează doar sălile care au înregistrat cel puțin un abonament nou în ultima lună.

```
--- 2. Recalculează și actualizează numărul de abonați pentru fiecare sală de fitness, numărând doar membrii distincti care au abonamente active cu data de expirare în viitor. Actualizează doar sălile care au înregistrat cel puțin un abonament nou în ultima lună.

✓ UPDATE SALA_FITNESS sf
SET nr_abonati =
  SELECT COUNT(DISTINCT a.id_membru)
    FROM ABONAMENT a
   WHERE a.id_sala = sf.id_sala
     AND a.activ = 1
     AND a.data_end >= SYSDATE
)
WHERE EXISTS (
  SELECT 1
    FROM ABONAMENT
   WHERE id_sala = sf.id_sala
     AND data_start > ADD_MONTHS(SYSDATE, -1)
);
✓ COMMIT;
```

```
[2025-05-15 21:02:52] completed in 2 ms
C##MIRUNA> UPDATE SALA_FITNESS sf
SET nr_abonati =
  SELECT COUNT(DISTINCT a.id_membru)
    FROM ABONAMENT a
   WHERE a.id_sala = sf.id_sala
     AND a.activ = 1
     AND a.data_end >= SYSDATE
)
WHERE EXISTS (
  SELECT 1
    FROM ABONAMENT
   WHERE id_sala = sf.id_sala
     AND data_start > ADD_MONTHS(SYSDATE, -1)
)
[2025-05-15 21:04:11] 2 rows affected in 6 ms
C##MIRUNA> COMMIT
[2025-05-15 21:04:11] completed in 2 ms
```

3. Actualizează prețul tuturor tipurilor de abonamente reducându-le cu 15% pentru sălile care au un echipament în stare defectă (nefuncțional) și cu 10% pentru sălile care au cel puțin un echipament cu o defecțiune minoră (defecțiune minoră).

```
✓ UPDATE TIP_ABONAMENT_SALA tas
SET pret = pret * 0.85
WHERE id_sala IN (
    SELECT e.id_sala
    FROM ECHIPAMENT e
    JOIN MENTENANTA_ECHIPAMENT me ON e.id_echipament = me.id_echipament
    WHERE LOWER(me.stare_verificare) = 'nefunctional'
);

✓ UPDATE TIP_ABONAMENT_SALA tas
SET pret = pret * 0.90
WHERE id_sala IN (
    SELECT e.id_sala
    FROM ECHIPAMENT e
    JOIN MENTENANTA_ECHIPAMENT me ON e.id_echipament = me.id_echipament
    WHERE LOWER(me.stare_verificare) = 'defectiune minoră'
)
AND id_sala NOT IN (
    SELECT e.id_sala
    FROM ECHIPAMENT e
    JOIN MENTENANTA_ECHIPAMENT me ON e.id_echipament = me.id_echipament
    WHERE LOWER(me.stare_verificare) = 'nefunctional'
);

✓ COMMIT;
```

```
C##MIRUNA> COMMIT
[2025-05-15 21:04:11] completed in 2 ms
C##MIRUNA> UPDATE TIP_ABONAMENT_SALA tas
SET pret = pret * 0.85
WHERE id_sala IN (
    SELECT e.id_sala
    FROM ECHIPAMENT e
    JOIN MENTENANTA_ECHIPAMENT me ON e.id_echipament = me.id_echipament
    WHERE LOWER(me.stare_verificare) = 'nefunctional'
)
[2025-05-15 21:05:30] 5 rows affected in 10 ms
C##MIRUNA> UPDATE TIP_ABONAMENT_SALA tas
SET pret = pret * 0.90
WHERE id_sala IN (
    SELECT e.id_sala
    FROM ECHIPAMENT e
    JOIN MENTENANTA_ECHIPAMENT me ON e.id_echipament = me.id_echipament
    WHERE LOWER(me.stare_verificare) = 'defectiune minoră'
)
AND id_sala NOT IN (
    SELECT e.id_sala
    FROM ECHIPAMENT e
    JOIN MENTENANTA_ECHIPAMENT me ON e.id_echipament = me.id_echipament
    WHERE LOWER(me.stare_verificare) = 'nefunctional'
)
[2025-05-15 21:05:30] 4 rows affected in 12 ms
C##MIRUNA> COMMIT
[2025-05-15 21:05:30] completed in 3 ms
```

## Operațiile de DELETE:

1. Șterge înregistrările din tabela MEMBRU\_CLASA pentru membrii care nu au abonamente active și sunt înscriși la clase care au capacitatea de cel puțin două ori mai mare decât numărul mediu de participanți prezenti.

```
✓ DELETE FROM MEMBRU_CLASA mc
WHERE mc.id_membru IN (
    SELECT m.id_membru
    FROM MEMBRU m
    WHERE NOT EXISTS (
        SELECT 1
        FROM ABONAMENT a
        WHERE a.id_membru = m.id_membru
        AND a.activ = 1
        AND a.data_end > SYSDATE
    )
)
AND mc.id_clasa IN (
    SELECT c.id_clasa
    FROM CLASA c
    WHERE c.capacitate_maxima > (
        SELECT COUNT(DISTINCT id_membru)
        FROM MEMBRU_CLASA
        WHERE id_clasa = c.id_clasa
        AND status_participare = 'Activ'
    ) * 2
);
✓ COMMIT;

[2025-05-15 21:05:30] completed in 3 ms
C##MIRUNA> DELETE FROM MEMBRU_CLASA mc
WHERE mc.id_membru IN (
    SELECT m.id_membru
    FROM MEMBRU m
    WHERE NOT EXISTS (
        SELECT 1
        FROM ABONAMENT a
        WHERE a.id_membru = m.id_membru
        AND a.activ = 1
        AND a.data_end > SYSDATE
    )
)
AND mc.id_clasa IN (
    SELECT c.id_clasa
    FROM CLASA c
    WHERE c.capacitate_maxima > (
        SELECT COUNT(DISTINCT id_membru)
        FROM MEMBRU_CLASA
        WHERE id_clasa = c.id_clasa
        AND status_participare = 'Activ'
    ) * 2
)
[2025-05-15 21:07:01] 6 rows affected in 25 ms
C##MIRUNA> COMMIT
[2025-05-15 21:07:01] completed in 2 ms
```

2. Șterge din tabela CLASA\_REZERVARE toate înregistrările pentru rezervări mai vechi de 3 luni sau sunt pentru clase conduse de traineri cu salariu sub 80% din media sălii respective.

```
✓ DELETE FROM CLASA_REZERVARE cr
  WHERE cr.id_rezervare IN (
    SELECT r.id_rezervare
      FROM REZERVARE r
     WHERE r.data_rezervare < ADD_MONTHS(SYSDATE, -3)
  )
  AND cr.id_clasa IN (
    SELECT c.id_clasa
      FROM CLASA c
     WHERE c.id_trainer IN (
       SELECT t.id_trainer
         FROM TRAINER t
        WHERE t.salariu < (
          SELECT AVG(salariu) * 0.8
            FROM TRAINER
           WHERE id_sala = t.id_sala
        )
      )
    );
✓ COMMIT;

--- 3. Șterge din tabelul MEMBRU_ECHIPAMENT toate înregistrările
  întocmite "Reducere" în plus cu echipamente trădătoare
```

```
)
```

[2025-05-15 21:07:01] 6 rows affected in 25 ms

C##MIRUNA> COMMIT

[2025-05-15 21:07:01] completed in 2 ms

C##MIRUNA> DELETE FROM CLASA\_REZERVARE cr
 WHERE cr.id\_rezervare IN (
 SELECT r.id\_rezervare
 FROM REZERVARE r
 WHERE r.data\_rezervare < ADD\_MONTHS(SYSDATE, -3)
 )
 AND cr.id\_clasa IN (
 SELECT c.id\_clasa
 FROM CLASA c
 WHERE c.id\_trainer IN (
 SELECT t.id\_trainer
 FROM TRAINER t
 WHERE t.salariu < (
 SELECT AVG(salariu) \* 0.8
 FROM TRAINER
 WHERE id\_sala = t.id\_sala
 )
 )
 )

3. Șterge din tabelul MEMBRU\_ECHIPAMENT toate înregistrările mai vechi de 2 luni și au folosit echipamente cu intensitate "Redusă". În plus, aceste echipamente trebuie: să fie declarate „nefuncționale” sau „cu defecțiuni” în ultima verificare.

```
0 ✓ DELETE FROM MEMBRU_ECHIPAMENT me
1   WHERE me.DATA_ULTIMA_ACTUALIZARE < ADD_MONTHS(SYSDATE, -2)
2     AND LOWER(me.intensitate) = 'redusă'
3     AND me.id_echipament IN (
4       SELECT e.id_echipament
5         FROM ECHIPAMENT e
6         WHERE EXISTS (
7           SELECT 1
8             FROM MENTENANTA_ECHIPAMENT m
9               WHERE m.id_echipament = e.id_echipament
10              AND LOWER(m.stare_verificare) IN ('nefunctional', 'defectiune')
11              AND m.data_verificare = (
12                SELECT MAX(m2.data_verificare)
13                  FROM MENTENANTA_ECHIPAMENT m2
14                    WHERE m2.id_echipament = e.id_echipament
15                  )
16                )
17      );
18
19 ✓ COMMIT;
```

```
)  
)  
[2025-05-15 21:07:59] completed in 13 ms  
C##MIRUNA> COMMIT  
[2025-05-15 21:07:59] completed in 1 ms  
C##MIRUNA> DELETE FROM MEMBRU_ECHIPAMENT me  
      WHERE me.DATA_ULTIMA_ACTUALIZARE < ADD_MONTHS(SYSDATE, -2)  
        AND LOWER(me.intensitate) = 'redusă'  
        AND me.id_echipament IN (  
          SELECT e.id_echipament  
            FROM ECHIPAMENT e  
            WHERE EXISTS (  
              SELECT 1  
                FROM MENTENANTA_ECHIPAMENT m  
                  WHERE m.id_echipament = e.id_echipament  
                    AND LOWER(m.stare_verificare) IN ('nefunctional', 'defectiune')  
                    AND m.data_verificare = (  
                      SELECT MAX(m2.data_verificare)  
                        FROM MENTENANTA_ECHIPAMENT m2  
                          WHERE m2.id_echipament = e.id_echipament  
                        )  
                      )  
                    )  
[2025-05-15 21:09:08] completed in 23 ms  
C##MIRUNA> COMMIT
```