



ACADEMIA DE STUDII ECONOMICE, BUCUREȘTI

FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ, SPECIALIZAREA
INFORMATICĂ ECONOMICĂ

PROIECT BAZE DE DATE

Nume: Dumitriu Ana Maria

Grupa: 1051

Serie: C

An universitar 2021 – 2022

Dumitriu Ana Maria

1051 C

Proiect BAZE DE DATE

Tema proiectului: **Sala de sport**

1. Descrierea temei (1/2 pag)

Proiectul reprezinta baza de date a unei sali de sport.

Exista mai multe sedii ale sali de sport, fiecare sediu avand denumire, capacitate si data de deschidere proprie. Fiecare sediu se afla la o adresa care contine oras, strada si numar. Fiecare sediu are unul sau mai multi angajati.

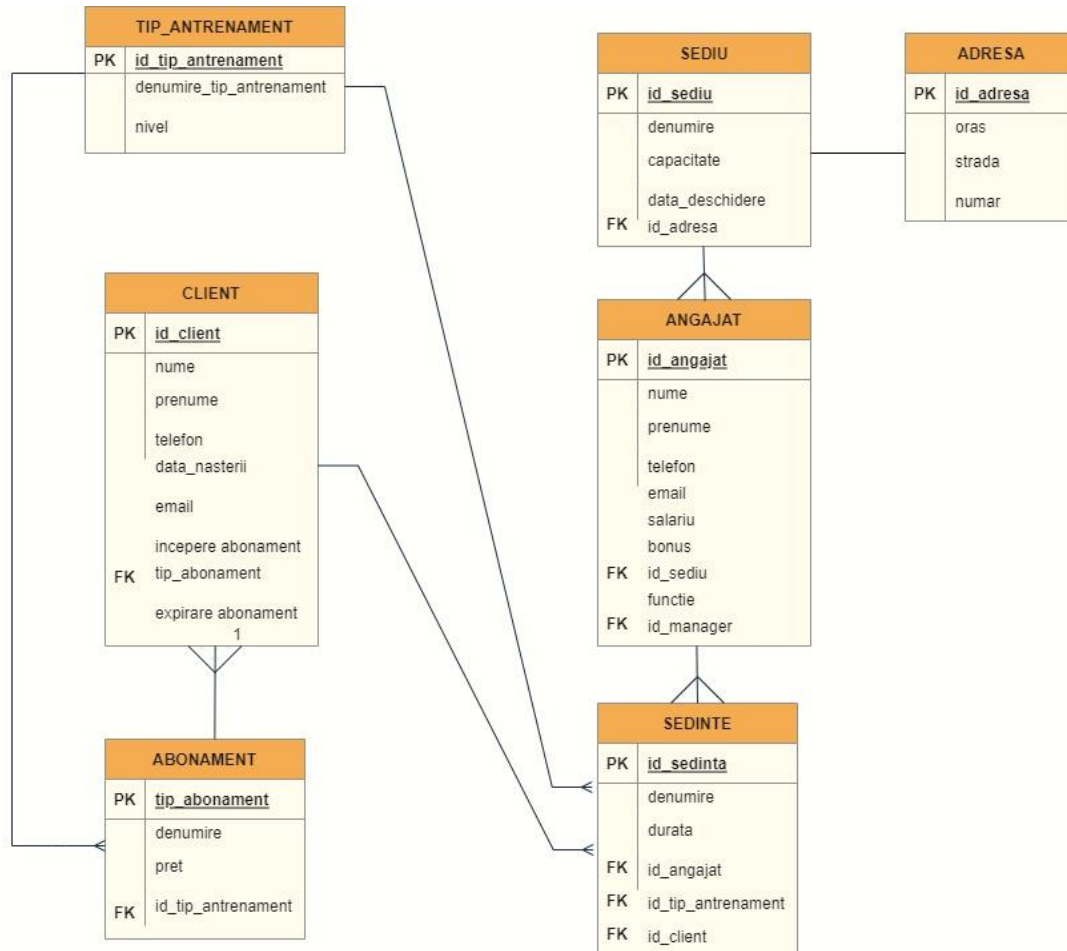
Pentru fiecare angajat stim numele, prenumele, telefonul, emailul, salariul, bonusul, sediul in care acesta lucreaza, functia acestuia si managerul care ii este superior. Fiecare angajat poate (daca acesta este antrenor) sa sustina una sau mai multe ore de antrenament (sedinte) impreuna cu un client.

Fiecare sedinta are un tip de antrenament (pentru fiecare tip de antrenament se cunosc denumirea care este alcatuita din initiala sportului practicat si numarul sedintei respectivului client si nivelul - un numar de la 1 la 5, reprezentand dificultatea.

Fiecare client participa la una sau mai multe sedinte. Despre fiecare client se cunosc urmatoarele: numele, prenumele, telefonul, data nasterii, emailul, data de incepere si data de expirare a abonamentului si tipul de abonament. Fiecare client are un abonament (iar un abonament poate fi avut de unul sau mai multi clienti).

Fiecare abonament are un tip de antrenament aferent un tip de antrenament (fiecare tip de antrenament face parte dintr-un abonament).

2. **Schema conceptuală** pentru modelarea temei alese. Din schema va trebui să rezulte tipul legăturilor dintre entități după modelul schemei de la seminar. Se poate realiza folosind orice instrument sau site (e.g., www.draw.io)



3. **Construirea bazei de date** – tabele (minim 4) și restricții de integritate. Exemplificarea operațiunile **LDD** (CREATE, ALTER, DROP) asupra tabelelor (min 7 instrucțiuni).

1. **Sa se creeze tabelele necesare:** CREATE (AD_ADRESE, AD_SEDII, AD_ANGAJATI, AD_TIP_ANTRENAMENTE, AD_ABONAMENTE, AD_CLIENTI, AD_SEDINTE).

```
CREATE TABLE AD_ADRESE
```

```
(
```

```
ID_ADRESA NUMBER(3) CONSTRAINT ID_ADRESA PRIMARY KEY,
```

```
ORAS VARCHAR2(50),
```

```
STRADA VARCHAR(50),
```

NUMAR VARCHAR2(5)

);

CREATE TABLE AD_SEDII

(ID_SEDIU NUMBER(3) CONSTRAINT SEDIU_ID PRIMARY KEY,

DENUMIRE VARCHAR2(50) NOT NULL,

CAPACITATE NUMBER(3) NOT NULL,

DATA_DESCHIDERE DATE NOT NULL,

ID_ADRESA NUMBER(3) NOT NULL,

CONSTRAINT FK_ADRESE FOREIGN KEY (ID_ADRESA) REFERENCES AD_ADRESE(ID_ADRESA)

);

CREATE TABLE AD_ANGAJATI

(

ID_ANGAJAT NUMBER(3) CONSTRAINT ID_ANG PRIMARY KEY,

NUME VARCHAR2(50) NOT NULL,

PRENUME VARCHAR2(50) NOT NULL,

TELEFON VARCHAR2(15) NOT NULL,

EMAIL VARCHAR2(50),

SALARIU FLOAT(5) NOT NULL,

BONUS FLOAT(5),

ID_SEDIU NUMBER(3) NOT NULL,

CONSTRAINT FK_SEDIU FOREIGN KEY (ID_SEDIU) REFERENCES AD_SEDII(ID_SEDIU)

);

CREATE TABLE AD_TIP_ANTRENAMENTE

(

ID_TIP_ANTRENAMENT NUMBER(3) CONSTRAINT ID_TIP_ANTRENAMENT PRIMARY KEY,

DENUMIRE_TIP_ANTRENAMENT VARCHAR2(50) NOT NULL,

NIVEL NUMBER(2) NOT NULL

);

CREATE TABLE AD_ABONAMENTE

```
(
TIP_ABONAMENT NUMBER(3) CONSTRAINT TIP_ABONAMENT PRIMARY KEY,
DENUMIRE VARCHAR2(50) NOT NULL,
PRET FLOAT(5),
ID_TIP_ANTRENAMENT NUMBER(3) NOT NULL,
CONSTRAINT FK_ID_TIP_ANTRENAMENT FOREIGN KEY (ID_TIP_ANTRENAMENT) REFERENCES
AD_TIP_ANTRENAMENTE(ID_TIP_ANTRENAMENT)
);
```

CREATE TABLE AD_CLIENTI

```
(
ID_CLIENT NUMBER(3) CONSTRAINT ID_CLIENT PRIMARY KEY,
NUME VARCHAR2(50) NOT NULL,
PRENUME VARCHAR2(50) NOT NULL,
TELEFON VARCHAR2(15) NOT NULL,
DATA_NASTERII DATE NOT NULL,
EMAIL VARCHAR2(50),
INCEPERE_ABONAMENT DATE NOT NULL,
EXPIRARE_ABONAMENT DATE NOT NULL,
TIP_ABONAMENT NUMBER(3),
CONSTRAINT FK_TIP_ABONAMENT FOREIGN KEY (TIP_ABONAMENT) REFERENCES
AD_ABONAMENTE(TIP_ABONAMENT)
);
```

CREATE TABLE AD_SEDINTE

```
(
ID_SEDINTA NUMBER(5) CONSTRAINT ID_SEDINTA PRIMARY KEY,
DENUMIRE VARCHAR2(50) NOT NULL,
DURATA NUMBER(5),
ID_ANGAJAT NUMBER(3),
ID_TIP_ANTRENAMENT NUMBER(3),
ID_CLIENT NUMBER(3),
CONSTRAINT FK_ID_ANG FOREIGN KEY (ID_ANGAJAT) REFERENCES
AD_ANGAJATI(ID_ANGAJAT),
```

```
CONSTRAINT FK_ID_TIP_ANTREN FOREIGN KEY (ID_TIP_ANTRENAMENT) REFERENCES  
AD_TIP_ANTRENAMENTE(ID_TIP_ANTRENAMENT),  
CONSTRAINT ID_CL FOREIGN KEY (ID_CLIENT) REFERENCES AD_CLIENTI(ID_CLIENT)  
);
```

```
CREATE TABLE AD_AGENDA  
(  
ID_CONTACT NUMBER(3) CONSTRAINT ID_CONTACT PRIMARY KEY,  
NUME VARCHAR2(50),  
NUMAR VARCHAR2(50)  
);
```

Utilizare ALTER si DROP

--2. Sa se starga coloana pret din sedinte si sa se adauge coloana PRET in tabela AD_SEDINTE, sa se adauge o **cheie de tip check** astfel incat pretul sa fie pozitiv.

```
ALTER TABLE AD_SEDINTE  
DROP COLUMN PRET;
```

```
ALTER TABLE AD_SEDINTE  
ADD (PRET FLOAT(5)NOT NULL);
```

```
ALTER TABLE AD_SEDINTE  
ADD CONSTRAINT SEDINTE_PRET_CK  
CHECK(PRET>=0);
```

--3. Sa se adauge o coloana care sa reprezinte data de **expirare** a unui abonament si o **restrictie** pentru tabela client, astfel incat data de expirare a unui abonament sa fie mai mare decat data inceperii respectivului abonament.

```
ALTER TABLE AD_CLIENTI  
ADD EXPIRARE_ABONAMENT DATE;
```

```
ALTER TABLE AD_CLIENTI  
ADD CONSTRAINT DURATA_ABONAMENT_CK  
CHECK(EXPIRARE_ABONAMENT>INCEPERE_ABONAMENT);
```

-- 4. Sa se **adauge o cheie unica** tabelii ad_sedii, astfel incat in tabela sa se regaseasca doar **combinatii unice** intre denumire si id_sediu (sa nu existe doua sedii cu aceeasi denumire)

```
ALTER TABLE AD_SEDII ADD CONSTRAINT ID_SEDIU_DENUMIRE_UK UNIQUE(ID_SEDIU, DENUMIRE);
```

-- 5. Sa se modifice tabela ad_tip_antrenamente astfel incat valorile campului nivel sa se regaseasca in **intervalul 1-5**

```
ALTER TABLE AD_TIP_ANTRENAMENTE  
ADD CONSTRAINT NIVEL_CK  
CHECK(NIVEL>=0 AND NIVEL<=5);
```

--6. Sa se **modifice** tabela ad_sedinte astfel incat durata unei sedinte sa nu poata avea valoarea **null**

```
ALTER TABLE AD_SEDINTE  
MODIFY(DURATA NOT NULL);
```

-- 7. Sa se **modifice tabela** ad_adrese astfel incat **valorile** din aceasta sa nu poata lua valoarea **null**

```
ALTER TABLE AD_ADRESE  
MODIFY(ORAS NOT NULL, STRADA NOT NULL, NUMAR NOT NULL);
```

--8. Sa se **adauge coloana** functie si coloana manager in tabela ad_angajati

```
ALTER TABLE AD_ANGAJATI  
ADD FUNCTIE VARCHAR2(50) NOT NULL;
```

```
ALTER TABLE AD_ANGAJATI  
ADD ID_MANAGER NUMBER(3);
```

4. Exemple cu operații de actualizare a datelor: INSERT, UPDATE, DELETE, MERGE (min 10). Obligatoriu, într-o tabelă trebuie să existe o înregistrare (rând) cu **numele studentului**, se va prezenta un printscreen după interogarea care demonstrează acest lucru.

--1. INSERAREA DATELOR IN TABELE:

```
INSERT INTO AD_ADRESE VALUES(1, 'Bucuresti', 'Poiana Florilor', 12);
```

```
INSERT INTO AD_ADRESE VALUES(2, 'Bucuresti', 'Moinesti', 8);
```

INSERT INTO AD_ADRESE VALUES(3, 'Bucuresti', 'Apusului', 5);

INSERT INTO AD_SEDII VALUES(10, 'BestFitness', 50, TO_DATE('10-MAR-2017', 'DD-MON-YYYY'), 1);

INSERT INTO AD_SEDII VALUES(11, 'FitGym', 30, TO_DATE('18-NOV-2018', 'DD-MON-YYYY'), 2);

INSERT INTO AD_SEDII VALUES(12, 'ZumbaGym', 20, TO_DATE('20-DEC-2020', 'DD-MON-YYYY'), 3);

INSERT INTO AD_ANGAJATI VALUES(10, 'Dumitriu', 'Ana Maria', '+40727274214', 'anadumitriu@gmail.com', 4200, 0.4, 10, 'Manager', null);

INSERT INTO AD_ANGAJATI VALUES(11, 'Badescu', 'Adela', '+40734274216', 'badescuadela@gmail.com', 3600, null, 10, 'Antrenor', 20);

INSERT INTO AD_ANGAJATI VALUES(12, 'Barbu', 'Maria', '+40726677818', 'barbumaria@gmail.com', 3500, 0.2, 10, 'Antrenor', 20);

INSERT INTO AD_ANGAJATI VALUES(13, 'Argesanu', 'Anca', '+40756274589', 'ancaargesanu@gmail.com', 2800, null, 11, 'Contabil', 10);

INSERT INTO AD_ANGAJATI VALUES(14, 'Macra', 'George', '+40728874280', 'georgemarca@gmail.com', 3900, 0.2, 10, 'Antrenor', 20);

INSERT INTO AD_ANGAJATI VALUES(15, 'Median', 'Sergiu', '+40728875674', 'mediansergiu@gmail.com', 2700, null, 11, 'Receptionist', 10);

INSERT INTO AD_ANGAJATI VALUES(16, 'Micsa', 'Radu', '+40725783480', 'micsaradu@gmail.com', 3200, 0.1, 11, 'Antrenor', 20);

INSERT INTO AD_ANGAJATI VALUES(17, 'Melencu', 'Sofia', '+40723365980', 'sofiamelencu@gmail.com', 2500, 0.2, 12, 'Paznic', 10);

INSERT INTO AD_ANGAJATI VALUES(18, 'Padurean', 'Silvia', '+40728833356', 'silviapadurean@gmail.com', 3450, 0.1, 12, 'Antrenor', 20);

INSERT INTO AD_ANGAJATI VALUES(19, 'Picu', 'Simona', '+40728823423', 'simonapicu@gmail.com', 2600, null, 12, 'Paznic', 10);

INSERT INTO AD_ANGAJATI VALUES(20, 'Mircea', 'Andrei', '+40788878356', 'mirceaandrei@gmail.com', 4000, 0.2, 10, 'Antrenor', 10);

INSERT INTO AD_TIP_ANTRENAMENTE VALUES(1, 'Cardio', 3);

INSERT INTO AD_TIP_ANTRENAMENTE VALUES(2, 'Zumba', 2);

INSERT INTO AD_TIP_ANTRENAMENTE VALUES(3, 'Forta', 5);

INSERT INTO AD_TIP_ANTRENAMENTE VALUES(4, 'Aerobic', 2);

INSERT INTO AD_TIP_ANTRENAMENTE VALUES(5, 'Gimnastica', 4);

INSERT INTO AD_TIP_ANTRENAMENTE VALUES(6, 'yoga', 1);

INSERT INTO AD_ABONAMENTE VALUES(21, 'C', 150, 1);

INSERT INTO AD_ABONAMENTE VALUES(22, 'Z', 180, 2);


```
INSERT INTO AD_ABONAMENTE VALUES(23, 'F', 130, 3);
```

```
INSERT INTO AD_ABONAMENTE VALUES(24, 'A', 120, 4);
```

```
INSERT INTO AD_ABONAMENTE VALUES(25, 'G', 160, 5);
```

```
INSERT INTO AD_ABONAMENTE VALUES(26, 'Y', 130, 6);
```

```
INSERT INTO AD_CLIENTI VALUES(101, 'Catinca', 'Rebic', '+40733823566', TO_DATE('26-FEB-1997', 'DD-MON-YYYY'), 'catincarebic@gmail.com', TO_DATE('23-MAR-2020', 'DD-MON-YYYY'), 22, TO_DATE('23-APR-2020', 'DD-MON-YYYY'));
```

```
INSERT INTO AD_CLIENTI VALUES(102, 'Diana', 'Riscov', '+40733823899', TO_DATE('11-MAR-1995', 'DD-MON-YYYY'), 'dianars@gmail.com', TO_DATE('10-NOV-2021', 'DD-MON-YYYY'), 21, TO_DATE('20-DEC-2021', 'DD-MON-YYYY'));
```

```
INSERT INTO AD_CLIENTI VALUES(103, 'Ioana', 'Echim', '+40733844657', TO_DATE('15-DEC-1999', 'DD-MON-YYYY'), 'echimioana@gmail.com', TO_DATE('08-APR-2021', 'DD-MON-YYYY'), 21, TO_DATE('15-MAY-2021', 'DD-MON-YYYY'));
```

```
INSERT INTO AD_CLIENTI VALUES(104, 'Stefan', 'Efimie', '+40733828883', TO_DATE('29-SEP-2001', 'DD-MON-YYYY'), 'stefanefimie@gmail.com', TO_DATE('18-FEB-2020', 'DD-MON-YYYY'), 26, TO_DATE('30-APR-2020', 'DD-MON-YYYY'));
```

```
INSERT INTO AD_CLIENTI VALUES(105, 'Catalina', 'Iordache', '+40734562314', TO_DATE('20-MAR-2002', 'DD-MON-YYYY'), 'iordachecatalina@gmail.com', TO_DATE('20-NOV-2019', 'DD-MON-YYYY'), null, TO_DATE('23-DEC-2019', 'DD-MON-YYYY'));
```

```
INSERT INTO AD_SEDINTE VALUES(70, 'C7', 30, 11, 1, 102);
```

```
INSERT INTO AD_SEDINTE VALUES(71, 'C8', 30, 11, 1, 102);
```

```
INSERT INTO AD_SEDINTE VALUES(72, 'Z3', 60, 16, 2, 101);
```

```
INSERT INTO AD_SEDINTE VALUES(73, 'Z4', 60, 16, 2, 101);
```

```
INSERT INTO AD_SEDINTE VALUES(74, 'C2', 30, 11, 1, 103);
```

```
INSERT INTO AD_SEDINTE VALUES(75, 'Y1', 90, 18, 6, 104);
```

```
INSERT INTO AD_SEDINTE VALUES(76, 'Y1', 90, 18, 6, 104);
```

```
INSERT INTO AD_SEDINTE VALUES(77, 'Y1', 90, 18, 6, 104);
```

-- 2. Sa se scrie o interogare care sa demonstreze faptul ca **numele** numele meu este prezent intr-un rand (inregistrare)

```
SELECT ID_ANGAJAT, NUME, PRENUME, TELEFON, EMAIL, SALARIU, BONUS, ID_SEDIU, FUNCTIE, ID_MANAGER, SALARIU*(1+BONUS) AS VENIT
```

```
FROM AD_ANGAJATI
```

```
WHERE LOWER(NUME)='dumitriu';
```

The screenshot shows a SQL worksheet with a query and its results. The query is:

```
SELECT ID_ANGAJAT, NUME, PRENUME, TELEFON, EMAIL, SALARIU, BONUS, ID_SEDIU, FUNCTIE, ID_MANAGER, SALARIU*(1+BONUS) AS VENIT
FROM AD_ANGAJATI
WHERE LOWER(NUME)='dumitriu';
```

The results are displayed in a table with the following columns: ID_ANGAJAT, NUME, PRENUME, TELEFON, EMAIL, SALARIU, BONUS, ID_SEDIU, FUNCTIE, ID_MANAGER, and VENIT. The results show one row for the employee with ID 10, named Dumitriu Ana Maria.

ID_ANGAJAT	NUME	PRENUME	TELEFON	EMAIL	SALARIU	BONUS	ID_SEDIU	FUNCTIE	ID_MANAGER	VENIT
10	Dumitriu	Ana Maria	+40727274214	anadumitriu@gmail.com	4200	0.4	10	Manager	(null)	5880

--3. Sa se **modifice bonusul** primit de angajatul cu numele Barbu si prenumele Maria in 0.3

```
UPDATE AD_ANGAJATI SET BONUS=0.3
WHERE NUME='Barbu' AND PRENUME='Maria';
```

--4. Sa se **modifice salariul** angajatilor receptionisti cu o crestere de 15%

```
UPDATE AD_ANGAJATI SET SALARIU=1.15*SALARIU
WHERE FUNCTIE='Receptionist';
```

--5. Sa se **modifice pretul** abonamentului cu id-ul 22, aplicandu-se o crestere cu 20%

```
UPDATE AD_ABONAMENTE SET PRET=0.8*PRET
WHERE TIP_ABONAMENT=22;
```

--6. Sa se **modifice bonusul** cu o crestere de 0.1 pentru toti angajatii care au bonus si acesta este mai mic sau egal cu 0.2

```
UPDATE AD_ANGAJATI SET BONUS=BONUS+0.1
WHERE BONUS IS NOT NULL AND BONUS<=0.2;
```

--7. Sa se **stearga** angajatul cu id-ul 19

```
DELETE FROM AD_ANGAJATI
WHERE ID_ANGAJAT=19;
```

--8. Sa se actualizeze tabela ad_agenda astfel incat numerele de telefon aferente id-ului clientului sa fie preluat din tabela clienti atunci cand avem deja in agenda numele si sa se insereze toate datele necesare in tabelul ad_agenda atunci cand nu avem clientul in ad_agenda (**MERGE**)

```
MERGE INTO AD_AGENDA AG
USING AD_CLIENTI C
ON (C.ID_CLIENT=AG.ID_CONTACT)
WHEN MATCHED THEN
UPDATE SET AG.NUMAR=C.TELEFON
WHEN NOT MATCHED THEN
INSERT (ID_CONTACT, NUME, NUMAR)
VALUES(C.ID_CLIENT, C.NUME, C.TELEFON);
```

5. Exemple de interogări cât mai variate și relevante pentru tema aleasă (min 15) care să combine următoarele elemente (preferabil toate elementele):

- -->, =, <=, !=, **IS NULL**, **LIKE**, **IN**, **BETWEEN**;

--1. Sa se afiseze toate sediile aflate in Bucuresti (folosind **LIKE**)

```
SELECT S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE, A.ORAS, A.STRADA
FROM AD_ADRESE A JOIN AD_SEDII S ON S.ID_ADRESA=A.ID_ADRESA
WHERE A.ORAS LIKE 'Buc%';
```

-- 2. Sa se afiseze toti clientii care **nu au un tip** de abonament

```
SELECT NUME, PRENUME, TELEFON, EMAIL
FROM AD_CLIENTI
WHERE TIP_ABONAMENT IS NULL;
```

-- 3. Sa se afiseze toti angajatii **mai putin** cel din functia de manager si cei din functia de contabil care au bonus **intre** 0.2 si 0.5

```
SELECT ID_ANGAJAT, NUME, PRENUME, TELEFON, EMAIL, BONUS, FUNCTIE
FROM AD_ANGAJATI
WHERE FUNCTIE NOT IN('Manager', 'Contabil') AND (BONUS BETWEEN 0.2 AND 0.5);
```

The screenshot shows an SQL Worksheet with a query and its results. The query is:

```
SELECT ID_ANGAJAT, NUME, PRENUME, TELEFON, EMAIL, BONUS, FUNCTIE
FROM AD_ANGAJATI
WHERE FUNCTIE NOT IN('Manager', 'Contabil') AND (BONUS BETWEEN 0.2 AND 0.5);
```

The results are displayed in a table with 7 columns: ID_ANGAJAT, NUME, PRENUME, TELEFON, EMAIL, BONUS, and FUNCTIE. There are 6 rows of data.

ID_ANGAJAT	NUME	PRENUME	TELEFON	EMAIL	BONUS	FUNCTIE
1	12 Barbu	Maria	+40726677818	barbumaria@gmail.com	0.3	Antrenor
2	14 Macra	George	+40728874280	georgemarca@gmail.com	0.3	Antrenor
3	16 Micsa	Radu	+40725783480	micsaradu@gmail.com	0.3	Antrenor
4	17 Melencu	Sofia	+40723365980	sofiamelencu@gmail.com	0.3	Paznic
5	18 Padurean	Silvia	+40728833356	silviapadurean@gmail.com	0.3	Antrenor
6	20 Mircea	Andrei	+40788878356	mirceaandrei@gmail.com	0.3	Antrenor

- JONCTIUNI (inner, outer); Utilizarea funcțiilor de grup, GROUP BY, HAVING;
- Utilizarea operatorilor UNION, MINUS, INTERSECT;

-- 4. Sa se afizeze sedintele facute de fiecare angajat (inner join)

```
SELECT A.NUME, A.PRENUME, A.FUNCTIE, S.DENUMIRE, S.DURATA
FROM AD_ANGAJATI A JOIN AD_SEDINTE S ON A.ID_ANGAJAT=S.ID_ANGAJAT;
```

--5. Sa se afizeze numele angajatilor care nu sustin sedinte (folosind minus)

```
SELECT A.ID_ANGAJAT, A.NUME, A.PRENUME, A.FUNCTIE
FROM AD_ANGAJATI A FULL OUTER JOIN AD_SEDINTE S ON A.ID_ANGAJAT=S.ID_ANGAJAT
GROUP BY A.ID_ANGAJAT, A.NUME, A.PRENUME, A.FUNCTIE
MINUS
SELECT A.ID_ANGAJAT, A.NUME, A.PRENUME, A.FUNCTIE
FROM AD_ANGAJATI A RIGHT JOIN AD_SEDINTE S ON S.ID_ANGAJAT=A.ID_ANGAJAT
GROUP BY A.ID_ANGAJAT, A.NUME, A.PRENUME, A.FUNCTIE;
```

--6. Sa se afiseze clientii care au abonament de **tip** cardio si abonament valid in data de 1 mai 2021 (folosind **intersect**)

```
SELECT C.ID_CLIENT, C.NUME, C.PRENUME, C.INCEPERE_ABONAMENT, A.TIP_ABONAMENT,
A.DENUMIRE, A.PRET
```

```
FROM AD_CLIENTI C JOIN AD_ABONAMENTE A ON C.TIP_ABONAMENT=A.TIP_ABONAMENT
```

```
WHERE UPPER(A.DENUMIRE)='C'
```

```
INTERSECT
```

```
SELECT C.ID_CLIENT, C.NUME, C.PRENUME, C.INCEPERE_ABONAMENT, A.TIP_ABONAMENT,
A.DENUMIRE, A.PRET
```

```
FROM AD_CLIENTI C JOIN AD_ABONAMENTE A ON C.TIP_ABONAMENT=A.TIP_ABONAMENT
```

```
WHERE C.EXPIRARE_ABONAMENT> TO_DATE('01-MAY-2021', 'DD-MON-YYYY');
```

--7. Sa se afiseze sediile care au mai mult de 2 angajati si **numarul de angajati** aferent

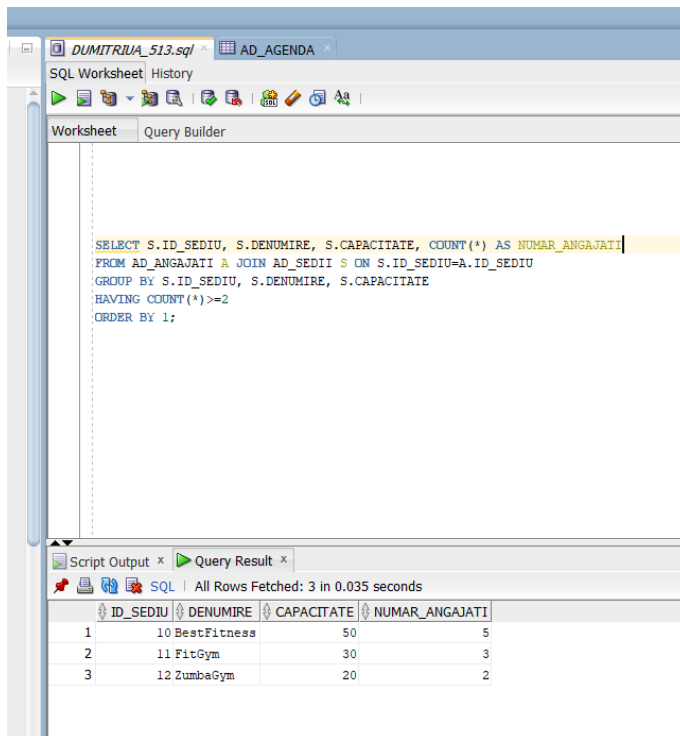
```
SELECT S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE, COUNT(*) AS NUMAR_ANGAJATI
```

```
FROM AD_ANGAJATI A JOIN AD_SEDII S ON S.ID_SEDIU=A.ID_SEDIU
```

```
GROUP BY S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE
```

```
HAVING COUNT(*)>=2
```

```
ORDER BY 1;
```



The screenshot shows a SQL IDE window with a query editor and a results pane. The query editor contains the following SQL code:

```
SELECT S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE, COUNT(*) AS NUMAR_ANGAJATI
FROM AD_ANGAJATI A JOIN AD_SEDII S ON S.ID_SEDIU=A.ID_SEDIU
GROUP BY S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE
HAVING COUNT(*)>=2
ORDER BY 1;
```

The results pane shows the output of the query, which is a table with 4 columns: ID_SEDIU, DENUMIRE, CAPACITATE, and NUMAR_ANGAJATI. The table contains 3 rows of data:

ID_SEDIU	DENUMIRE	CAPACITATE	NUMAR_ANGAJATI
10	BestFitness	50	5
11	FitGym	30	3
12	ZumbaGym	20	2

--8.Sa se afiseze toti angajatii care nu au functie de antrenor si care nu au bonus (folosind **union**)

```
SELECT * FROM AD_ANGAJATI
WHERE UPPER(FUNCTIE)!='ANTRENOR'
UNION
```

```
SELECT * FROM AD_ANGAJATI
WHERE BONUS IS NULL;
```

• Utilizarea funcțiilor la nivel de rând (obligatoriu: TO_CHAR, TO_DATE, EXTRACT, SUBSTR, SYSDATE, DECODE, CASE, NVL);

--9. Sa se afiseze clientii care si-au inceput abonamentele intre luna martie anul 2021 si [data curenta](#)

```
SELECT ID_CLIENT, NUME, PRENUME, TIP_ABONAMENT, INCEPERE_ABONAMENT
FROM AD_CLIENTI
WHERE INCEPERE_ABONAMENT BETWEEN TO_DATE('2021-03-01', 'YYYY-MM-DD') AND SYSDATE;
```

--10. Sa se afiseze toate abonamentele din [anul trecut](#) (2020)

```
SELECT ID_CLIENT, NUME, PRENUME, TIP_ABONAMENT, INCEPERE_ABONAMENT
FROM AD_CLIENTI
WHERE TO_CHAR(EXTRACT(YEAR FROM INCEPERE_ABONAMENT))=TO_CHAR(EXTRACT(YEAR
FROM SYSDATE)-1);
```

--11. Sa se afiseze sediile de pe strada [Moinesti](#)

```
SELECT S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE, A.ORAS || ', ' || A.STRADA AS ADRESA_COMPLETA
FROM AD_ADRESE A JOIN AD_SEDII S ON S.ID_ADRESA=A.ID_ADRESA
WHERE UPPER(SUBSTR(A.ORAS || ', ' || A.STRADA, 11)) LIKE '%MOINESTI%';
```

--12. Precizati care sunt tipurile de abonamente, folosind [DECODE](#)

```
SELECT TIP_ABONAMENT, DENUMIRE, PRET,
DECODE(DENUMIRE, 'C', 'Cardio', 'Z', 'Zumba', 'F', 'Forta', 'A', 'Aerobic', 'G', 'Gimnastica', 'Y', 'Yoga', 'Aletle')
AS TIP_SPORT
FROM AD_ABONAMENTE
ORDER BY 1;
```

--13. Sa se afiseze [venitul](#) lunar al unui angajat, incluzand bonusul cand acesta exista

```
SELECT ID_ANGAJAT, NUME, PRENUME, SALARIU, NVL(BONUS, 0) BONUS, SALARIU+
SALARIU*NVL(BONUS, 0) VENIT
```

```
FROM AD_ANGAJATI  
ORDER BY VENIT;
```

- Utilizarea lui CASE;

-- 14. Sa se afiseze tipul de antrenament, denumirea si nivelul acestuia, utilizandu-se CASE pentru afisarea nivelului

```
SELECT ID_TIP_ANTRENAMENT, DENUMIRE_TIP_ANTRENAMENT, NIVEL,  
CASE WHEN NIVEL=1 OR NIVEL=2 THEN 'Nivel de dificultate scazut'  
      WHEN NIVEL=3 OR NIVEL=4 THEN 'Nivel de dificultate mediu'  
      WHEN NIVEL>=5 THEN 'Nivel de dificultate ridicat'  
      ELSE 'Nivelul de dificultate nu este corect inregistrat'  
END AS NIVEL_DIFICULTATE  
FROM AD_TIP_ANTRENAMENTE;
```

- Subcereri simple și corelate;

--15. Sa se afiseze angajatii care au salariul mai mic decat media salariilor tuturor angajatilor

```
SELECT ID_ANGAJAT, NUME, PRENUME, SALARIU, NVL(BONUS, 0) BONUS, SALARIU+  
SALARIU*NVL(BONUS, 0) VENIT  
FROM AD_ANGAJATI  
WHERE SALARIU+ SALARIU*NVL(BONUS, 0)<(SELECT AVG(SALARIU+ SALARIU*NVL(BONUS, 0))  
FROM AD_ANGAJATI)  
ORDER BY VENIT DESC;
```

The screenshot shows an SQL Worksheet window with a query and its results. The query is as follows:

```
SELECT ID_ANGAJAT, NUME, PRENUME, SALARIU, NVL(BONUS, 0) BONUS, SALARIU+ SALARIU*NVL(BONUS, 0) VENIT
FROM AD_ANGAJATI
WHERE SALARIU+ SALARIU*NVL(BONUS, 0) < (SELECT AVG(SALARIU+ SALARIU*NVL(BONUS, 0)) FROM AD_ANGAJATI)
ORDER BY VENIT DESC;
```

The results are displayed in a table with the following data:

ID_ANGAJAT	NUME	PRENUME	SALARIU	BONUS	VENIT
1	16 Micsa	Radu	3200	0.3	4160
2	15 Median	Sergiu	4100	0	4100
3	11 Badescu	Adela	3600	0	3600
4	17 Melencu	Sofia	2500	0.3	3250
5	13 Argesanu	Anca	2800	0	2800

--16. Sa se afiseze antrenorii care au **cel mai mare numar** de sedinte efectuate cu clientii

```
SELECT ID_ANGAJAT, NUME, PRENUME, FUNCTIE
FROM AD_ANGAJATI
WHERE ID_ANGAJAT = (SELECT MAX(ID_ANGAJAT) FROM AD_SEDINTE)
ORDER BY 1;
```

• **CREATE/INSERT/UPDATE/DELETE + SELECT;**

--17. Sa se creeze o copie a angajatilor care sunt antrenori in tabela ad_antrenori (**create+select**)

```
CREATE TABLE AD_ANTRENORI
AS(SELECT * FROM AD_ANGAJATI
WHERE upper(FUNCTIE)='ANTRENOR');
```

• **Construirea și utilizarea altor obiecte ale bazei de date: tabele virtuale, indecși, sinonime și secvențe.**

--18. Sa se creeze o tabela virtuala (**view**) care sa contina datele despre sedii si despre adresele acestora

```
CREATE OR REPLACE VIEW VIEW_SEDII_ADRESE AS  
SELECT S.ID_SEDIU, S.DENUMIRE, S.CAPACITATE, S.DATA_DESCHIDERE, A.ORAS, A.STRADA,  
A.NUMAR  
FROM AD_SEDII S JOIN AD_ADRESE A ON A.ID_ADRESA=S.ID_ADRESA  
ORDER BY S.ID_SEDIU;
```

--19. Sa se creezed **index** pentru tabela ad_angajati, utilizandu-se numele de familie

```
CREATE INDEX idx_nume ON AD_ANGAJATI(INITCAP(NUME));
```

--20. Sa se creeze un **sinonim** pentru ad_adrese

```
--CREATE SYNONYM AD_Locatiile_Salilor FOR AD_ADRESE;
```

--21. Sa se creeze o **secventa** care sa asigure unicitatea id-ului din tabela clienti

```
CREATE SEQUENCE SEQ_CLIENTI  
START WITH 101 INCREMENT BY 1  
MAXVALUE 200 NOCYCLE;
```

• Cereri ierarhice (**CONNECT BY, PRIOR, LEVEL, SYS_CONNECT_BY_PATH**)

--22. Sa se afiseze subordonatii directi si indirecti ai angajatului cu id_angajat egal cu 10

```
SELECT ID_ANGAJAT, LPAD(' ',LEVEL*4-2)||NUME||' '||PRENUME ANGAJAT,  
SALARIU, BONUS, ID_MANAGER, LEVEL FROM AD_ANGAJATI  
CONNECT BY PRIOR ID_ANGAJAT=ID_MANAGER  
START WITH ID_ANGAJAT=10  
ORDER BY LEVEL;
```

DUMITRIUA_513.sql

SQL Worksheet History

Worksheet Query Builder

```
SELECT ID_ANGAJAT, LPAD(' ', LEVEL*4-2) || NUME || ' ' || PRENUME ANGAJAT,  
SALARIU, BONUS, ID_MANAGER, LEVEL FROM AD_ANGAJATI  
CONNECT BY PRIOR ID_ANGAJAT=ID_MANAGER  
START WITH ID_ANGAJAT=10  
ORDER BY LEVEL;
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.034 seconds

ID_ANGAJAT	ANGAJAT	SALARIU	BONUS	ID_MANAGER	LEVEL
1	10 Dumitriu Ana Maria	4200	0.4	(null)	1
2	15 Median Sergiu	4100	(null)	10	2
3	17 Melencu Sofia	2500	0.3	10	2
4	13 Argesanu Anca	2800	(null)	10	2
5	20 Mircea Andrei	4000	0.3	10	2
6	11 Badescu Adela	3600	(null)	20	3
7	12 Barbu Maria	3500	0.3	20	3
8	14 Macra George	3900	0.3	20	3
9	16 Micsa Radu	3200	0.3	20	3
10	18 Padurean Silvia	3500	0.3	20	3