

Gestiunea unui campionat

Modelul real

Baza de date cuprinde informatii referitoare la desfasurarea unor campionate tipice de ciclism. In acest scop se vor stoca date despre campionate, despre stagiile pe care le vor tine, disciplinele la care ciclistii vor putea concura, despre sponsori si legaturile lor de finantare.

Ciclistii vor putea participa la o superpromotie pe fiecare traseu in urma carora invingatorilor li se acorda puncta bonus.

Scopul bazei se rezuma la a putea organiza in decursul unei zile distributia ciclistilor la evenimentele stabilite de catre organizator.

DB-ul respecta o serie de restrictii firesti. Asadar, o grupa va fi alcatuita din oricat de multi ciclisti. Un ciclist se poate afla in oricate grupe, asa cum orice grupa poate participa la oricate discipline din cadrul unui campionat. Disciplinele pe de alta parte vor fi sustinute singular de catre fiecare competitie in parte, ceea ce inseamna ca odata ce o disciplina a fost organizata de un campionat, ea nu va mai putea exista si in urmatorul.

Totodata, sponsorii lucreaza atat cu campionatele, oferind finantare, cat si cu ciclisti in scopuri comerciale. Ei pot fi sponsorizati de oricat de multe companii se ofera.

In continuare, un ciclist care castiga o superpromotie, cu alte cuvinte o cursa specifica fiecarui traseu si care se parcurge o singura data, va primi punctaj bonus.

Impreuna cu medicii, grupa concurenta se asociaza unui traseu si se va forma orarul competitiv.

Reguli de functionare

Modelul de date prezinta urmatoarele restrictii de functionare:

1. O disciplina nu poate fi sustinuta decat de un campionat
2. O grupa are voie sa participe la oricate discipline doreste, precum ciclistii pot fi alocati in infinit de multe grupe
3. Un sponsor poate finanta oricat de multe campionate sau ciclisti simultan.
4. Un ciclist poate concura in mai multe superpromotii, dar numai o singura data.
5. O grupa are un singur medic responsabil per cursa.

Entitati

Urmatoarele structuri in modelul de date se numesc entitati: CAMPIONAT, DISCIPLINA, GRUPA, CICLIST, SPONSOR, MEDIC, TRASEU.

In cele ce urmeaza, voi descrie complet entitatile, precizant cheia primara. Entitati dependente sunt ALOCAT, FINANTARE, SPONSORIZARE, ORAR.

CAMPIONAT = eveniment unde pasionatii de ciclism se intrec pentru a demonstra performanta individuala. Cheia primara este id_camp.

DISCIPLINA = categorie de ciclism, unde participa persoane special antrenate si cu echipament specific. Cheia primara este id_disciplina.

GRUPA = asociatie de persoane, cu scop comun de regula stabilita in functie de performantele indivizilor ce o alcatuiesc. Cheia primara este id_grupa.

CICLIST = individ pasionat si antrenat pentru competitii sportive de ciclism, care doreste sa arate performanta sa intr-un eveniment organizat. Cheia primara este id_ciclist.

SPONSOR = companie cu posibilitati financiare care doreste sa isi promoveze imaginea in urma dotarii participantilor sau a unui campionat cu materiale sau bani. Cheia primara este id_sponsor.

MEDIC = o persoana educata sa ajute persoanele vatamate in urma ciclismului, sport deosebit de periculos cand este dus la extrem. Cheia primara este id_medic.

TRASEU = circuit, locatie pusa la dispozitie de catre autoritati pentru sportivi sa concureze. Cheia primara este id_traseu.

ALOCAT = modalitatea unui ciclist de a fi repartizat intr-o grupa. Cheia primara compusa este formata din id_ciclist si id_grupa.

FINANTARE = tine evidenta companiilor care ajuta direct campionatul. Cheia primara este compusa din id_camp si id_sponsor.

SPONSORIZARE = mijlocul prin care o companie sustine unul sau mai multi concurent cu particularitati comune. Cheia primara compusa este formata din id_ciclist si id_sponsor.

ORAR = programul unui campionat, sustinand repartitia traseelor, a medicilor si a grupelor pentru a desfasura o singura cursa, de fiecare data. Cheia primara compusa are id_grupa, id_medic, id_traseu.

Relatii

DISCIPLINA_sustinuta_de_GRUPA = relatie tip many-to-many care leaga entitatile GRUPA si DISCIPLINA, denuntand legatura dintre acestea. Cardinalitatea minima este 1:0 (o disciplina trebuie sa fie sustinuta de o grupa) si cardinalitatea maxima m:n (o disciplina poate fi sustinuta de indiferent de multe grupe).

GRUPA_formata_din_CICLIST = relatie de tip one to many; cardinalitatea minima este 1:1 (grupa trebuie sa aiba minim un ciclist) si cardinalitatea maxima m:n (oricate grupe pot avea oricati ciclisti).

SPONSOR_promoveaza_CICLIST = many-to-many; cardinalitatea minima 1:0 (nu este necesar ca un ciclist sa fie sponsorizat), iar cardinalitatea maxima m:n (un ciclist poate avea oricat de multi spnsori).

SPONSOR_finanteaza_CAMPIONAT = many-to-many; cardinalitatea minima este 1:0 si cardinalitatea maxima m:n.

CICLIST_superpromo_TRASEU = relatie one-to-many, care leaga CICLIST de TRASEU pentru a tine evidenta superpromotiilor. Cardinalitatea minima este 1:0 si cardinalitatea maxima este 1:m (un ciclist poate participa pe fiecare traseu la eveniment).

GRUPA_ajutat_MEDIC_la_loc_TRASEU = relatie tip 3 ce leaga grupa de medic si traseu. Denumirea este ORAR.

Atribute

Campionat

1. id_camp = integer, codul unui campionat PK
2. nume = sir caractere de lungime mai mica decat 30, numele unui campionat NOT NULL
3. an = integer, NOT NULL
4. mail = sir caractere de lungime mai mica decat 30

Disciplina

1. id_disciplina = integer PK, codul disciplinei in sistem
2. in_champ = integer FK, codul trebuie sa corespunda unei chei primare din campionat
3. nume = sir caractere de lungime mai mica decat 20, NOT NULL

Grupa

1. id_grupa = integer PK, codul unei grupe
2. in_discipline = integer FK, codul trebuie sa corespunda unei chei primare din disciplina.

Alocat

1. ciclist_ = integer PK, FK, codul trebuie sa corespunda unei chei primare din ciclist.
2. grupa_ = integer PK, FK, codul trebuie sa corespunda unei chei primare din grupa

Ciclist

1. id_ciclist = integer PK
2. nume = sir caractere < 15 NOT NULL
3. prenume = sir caractere < 15 NOT NULL
4. sex = integer, DEFAULT NULL
5. data_nastere = variabila tip data calendaristica, NOT NULL

6. telefon = sir caractere < 16, NOT NULL
7. punctaj = integer, DEFAULT 0

Sponsor

1. id_sponsor = integer PK,
2. nume = sir caractere < 25 NOT NULL
3. mainly_distributes = sir caractere < 15, NOT NULL, ofera informatie legat de principalul produs pe care firma il comercializeaza
4. website = sir caractere < 40 NOT NULL

Finantare

1. sponsor = integer, PK, FK codul trebuie sa corespunda cheii primare din sponsor
2. campionat = integer, PK, FK codul trebuie sa corespunda cheii primare din campionat

Promovari

1. ciclist = integer, PK, FK, codul trebuie sa corespunda cheii primare din ciclist
2. sponsor = integer PK, FK codul trebuie sa corespunda cheii primare din sponsor
3. data_contract = data calendaristica, NOT NULL

Medici

1. id_medic = integer, cheie primara
2. nume = sir de caractere <50, numele complet not null
3. varsta = integer, NOT NULL
4. exp_anterioara = BIT, default 0, memoreaza daca medicul a mai activat in campionate de ciclism sau este prima oara

Traseu

1. id_traseu = integer, PK
2. lungime = float, NOT NULL
3. tip = CHAR, clasificarea traseului dupa standardele internationale

Superpromotie

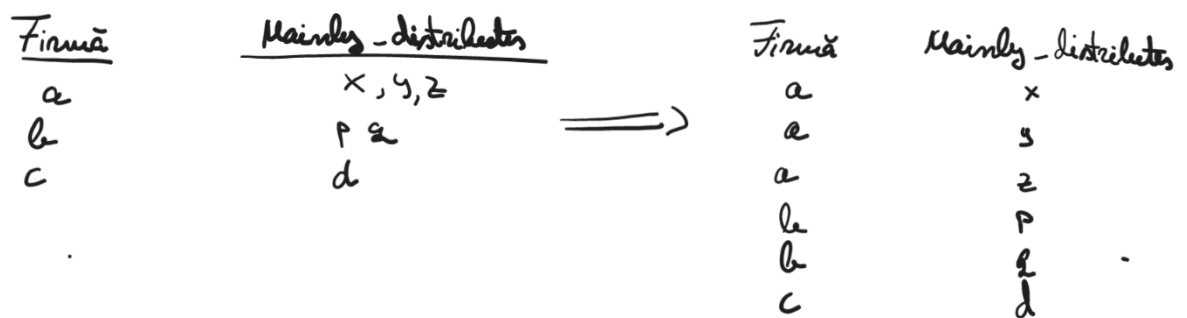
1. ciclist = integer, PK, FK, codul trebuie sa corespunda cheii primare din ciclist
2. traseu = integer, PK, FK, codul trebuie sa corespunda cheii primare din traseu
3. timp = integer, NOT NULL timpul pe care ciclistul l-a scos
4. castiga = BIT, DEFAULT 0, memoreaza daca superpromotia a fost castigata

Orar

1. grupa, PK, FK codul trebuie sa corespunda cheii primare din grupa
2. traseu, PK, FK codul trebuie sa corespunda cheii primare din traseu
3. medic, PK, FK codul trebuie sa corespunda cheii primare din medic

Normalizare

First normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if and only if no attribute domain has relations as elements.

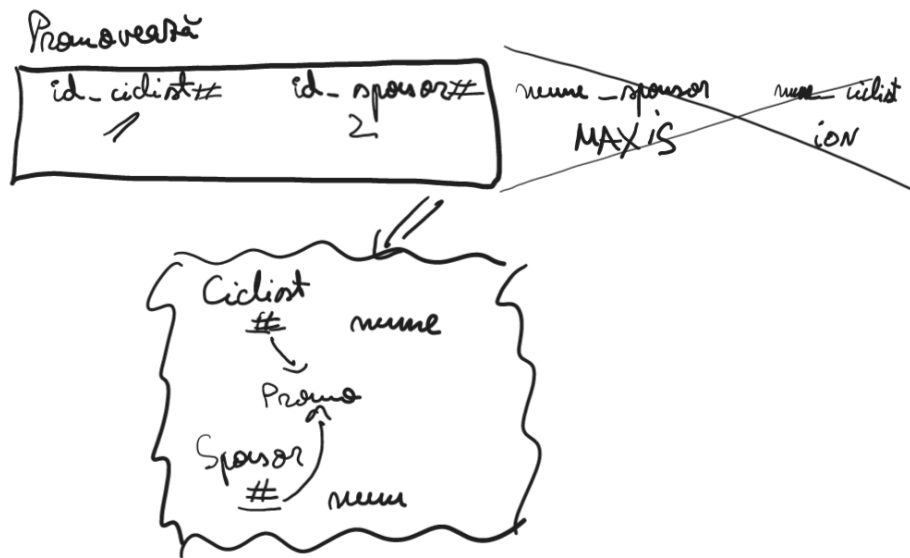


A relation is in the 2NF form if it fulfills the following two requirements:

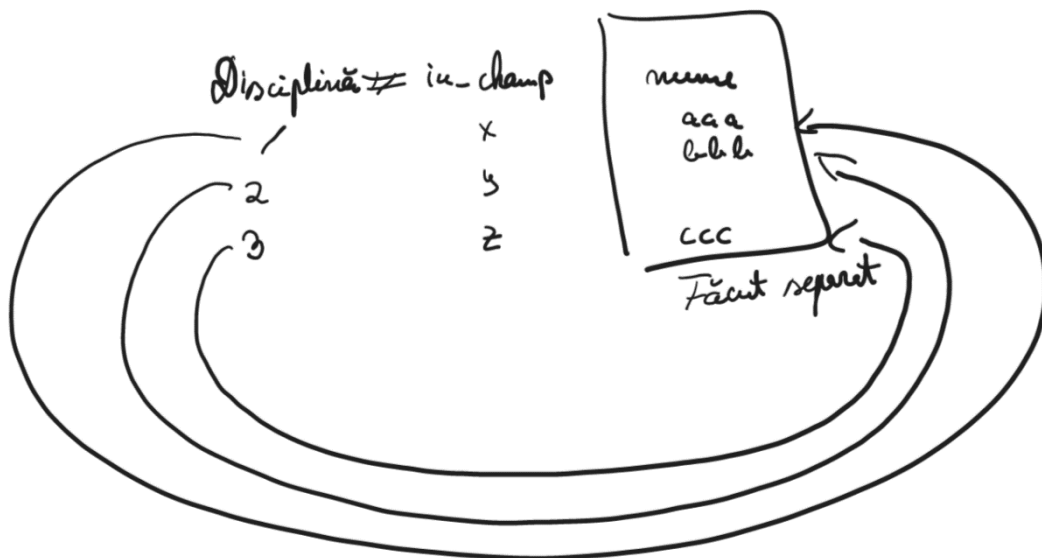
#It is in first normal form.

#It does not have any non-prime attribute that is functionally dependent on any proper subset of any candidate key of the relation. A non-prime attribute of a relation is an attribute that

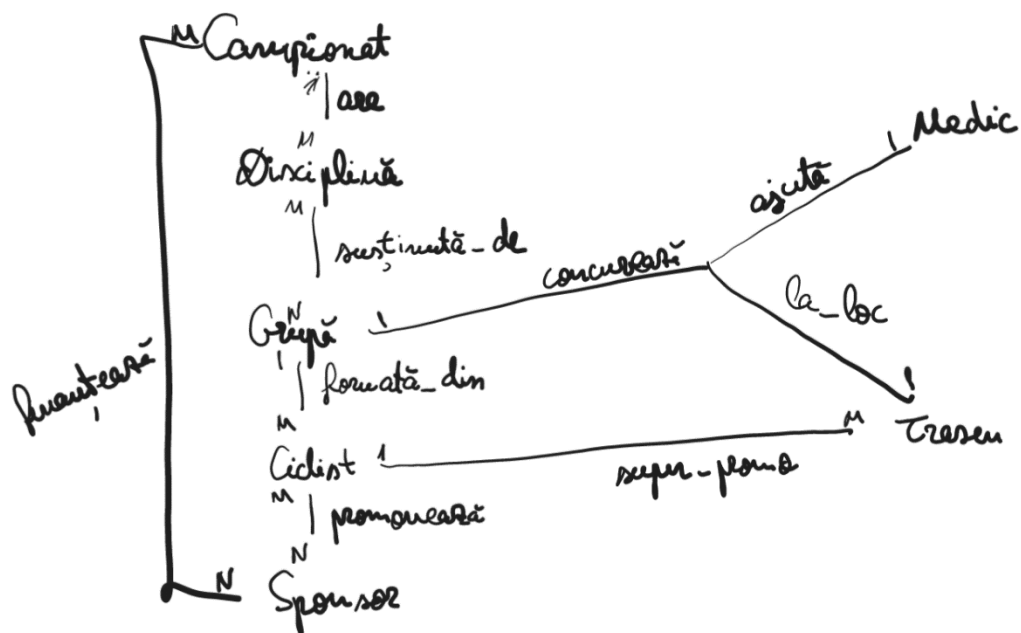
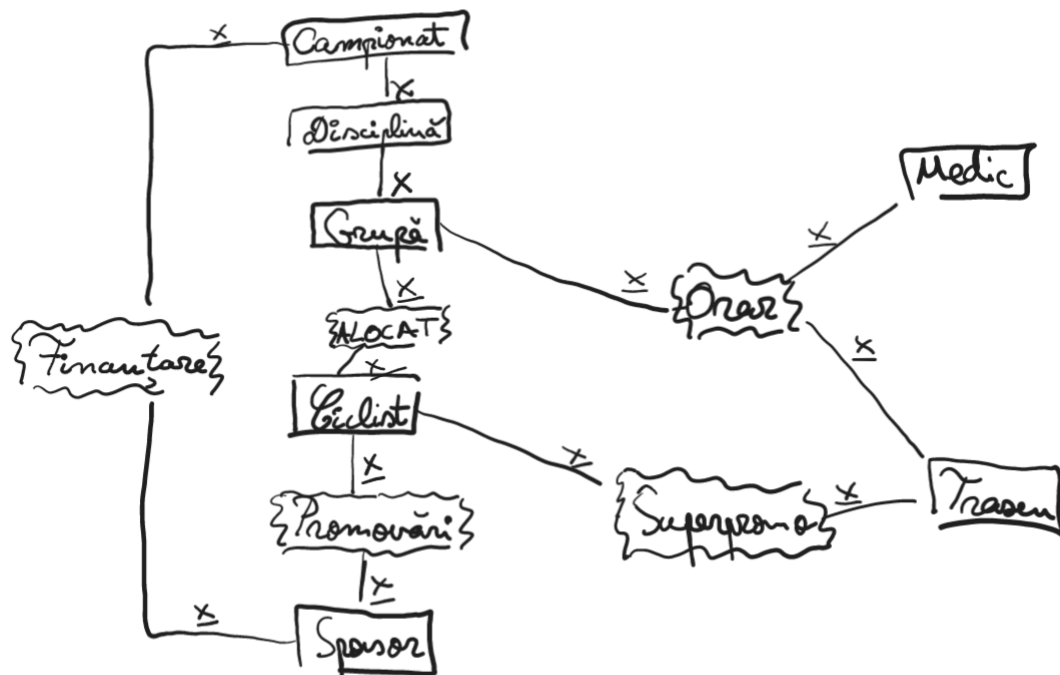
is not a part of any candidate key of the relation.



3NF: Codd defined this as a relation in second normal form where all non-prime attributes depend only on the candidate keys and do not have a transitive dependency on another key



Conceptual & ER/D



SQL

```
CREATE TABLE Campionat(
    id_camp INT PRIMARY KEY,
    nume VARCHAR(30) NOT NULL,
    an INT NOT NULL,
    mail VARCHAR(30)
);
SELECT * FROM Campionat;
INSERT INTO Campionat VALUES (1, 'C.N. Downhill', 2022, 'cheilegradistei@office.com');
INSERT INTO Campionat VALUES (2, 'Downhill Word Championship', 2022, 'worldcycling@managme');
INSERT INTO Campionat VALUES (3, 'Redbull Rampage', 2023, 'redbull@tv.com');
INSERT INTO Campionat VALUES (4, 'Tour de France', 2023, 'francecycliste@gmail.com');
INSERT INTO Campionat VALUES (5, 'Whistler Closing Week Showdown', 2022, 'whistler@office');
INSERT INTO Campionat VALUES (6, 'BikeXpert Challenge', 2022, 'bikexpert@team.ro');

CREATE TABLE Disciplina (
    id_disciplina INT PRIMARY KEY,
    in_champ INT FOREIGN KEY REFERENCES Campionat(id_camp),
    nume VARCHAR(20) NOT NULL
);

INSERT INTO Disciplina VALUES (132, 1, 'Downhill');
INSERT INTO Disciplina VALUES (133, 4, 'Road');
INSERT INTO Disciplina VALUES (134, 2, 'Gravity Downhill');
INSERT INTO Disciplina VALUES (135, 2, 'Pump Track');
INSERT INTO Disciplina VALUES (136, 2, 'FreeStyle');
```



```

CREATE TABLE Grupa(
  id_grupa INT PRIMARY KEY,
  in_discipline INT FOREIGN KEY REFERENCES Disciplina(id_disciplina)
);
INSERT INTO Grupa VALUES (11, 134);
INSERT INTO Grupa VALUES (12, 134);
INSERT INTO Grupa VALUES (13, 134);
INSERT INTO Grupa VALUES (21, 135);
INSERT INTO Grupa VALUES (31, 136);
INSERT INTO Grupa VALUES (49, 132);
INSERT INTO Grupa VALUES (50, 132);

```

```

DROP TABLE Alocat;
CREATE TABLE Alocat(
  ciclist_ INT,
  grupa_ INT,
  PRIMARY KEY (ciclist_, grupa_),
  FOREIGN KEY (ciclist_) REFERENCES Ciclist(id_ciclist),
  FOREIGN KEY (grupa_) REFERENCES Grupa(id_grupa)
)

INSERT INTO Alocat VALUES(480,11);
INSERT INTO Alocat VALUES(500,11);
INSERT INTO Alocat VALUES(520,11);
INSERT INTO Alocat VALUES(530,11);
INSERT INTO Alocat VALUES(570,12);
INSERT INTO Alocat VALUES(580,12);
INSERT INTO Alocat VALUES(590,12);
INSERT INTO Alocat VALUES(600,12);
INSERT INTO Alocat VALUES(490,13);
INSERT INTO Alocat VALUES(540,13);
INSERT INTO Alocat VALUES(550,13);
INSERT INTO Alocat VALUES(560,13);

```

```

CREATE TABLE Ciclist(
  id_ciclist INT PRIMARY KEY,
  nume VARCHAR(15) NOT NULL,
  prenume VARCHAR(15) NOT NULL,
  sex INT DEFAULT NULL,
  data_nasterii DATE NOT NULL,
  telefon VARCHAR(16) NOT NULL,
  punctaj INT DEFAULT 0
);

```

```

INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Remy','Metallier', 1,'1984-07-12','+81
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Vali','Holl',0,'1995-02-12','+13123214
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Troy','Brosnan',1,'1989-01-02','+21321
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Jordie','Lynn',1,'1987-11-11','+213129
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Corinne','Sutter',0,'2000-03-07','+312
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Nicole','Schimdhofen',0,'2001-09-11','
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Mihaela','Shiffrin',0,'1998-08-27','+4
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Brandon','Semenuk',1,'2001-01-16','+91
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Brett','Rheeder',1,'1995-12-29','+9473
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Tom','van Steenbergen',1,'1989-11-01',
INSERT INTO Ciclist VALUES(NEXT VALUE FOR SEQ_CICL,'Brendan','Fairclough',1,'1999-02-12','

```

```

-- Aratati tabelul ciclistilor

```

```


```

```

CREATE TABLE Sponsor(
  id_sponsor INT PRIMARY KEY,
  nume VARCHAR(25) NOT NULL,
  mainly_distributes VARCHAR(15) NOT NULL,
  website VARCHAR(40) NOT NULL
);

```

```

INSERT INTO Sponsor VALUES (13, 'NUKEPROOF', 'Pedale', 'nukeproof.com');
INSERT INTO Sponsor VALUES (14, 'MAXXIS', 'Anvelope', 'maxxis.com');
INSERT INTO Sponsor VALUES (15, 'Mucoff', 'Lubrifianti', 'mucoff.com');
INSERT INTO Sponsor VALUES (16, 'KMC', 'Lanturi', 'kmc-chains.org');
INSERT INTO Sponsor VALUES (17, 'WTB', 'Anvelope', 'wtbtyres.com');
INSERT INTO Sponsor VALUES (18, 'Cube', 'Cadre', 'cube.de');
INSERT INTO Sponsor VALUES (19, 'Shimano', 'Deraioare', 'shimano.eu');
INSERT INTO Sponsor VALUES (20, 'SRAM', 'Deraioare', 'sram.us');

```

```

CREATE TABLE Finantare(
  sponsor INT,
  campionat INT,
  PRIMARY KEY (sponsor, campionat),
  FOREIGN KEY (sponsor) REFERENCES Sponsor(id_sponsor),
  FOREIGN KEY (campionat) REFERENCES Campionat(id_camp)
);

```

```

INSERT INTO Finantare VALUES(15,2);
INSERT INTO Finantare VALUES(19,2);
INSERT INTO Finantare VALUES(14,2);
INSERT INTO Finantare VALUES(20,2);
INSERT INTO Finantare VALUES(15,1);

```

```
CREATE TABLE Promovari (  
    ciclist INT,  
    sponsor INT,  
    data_contract DATE NOT NULL,  
    PRIMARY KEY (ciclist, sponsor),  
    FOREIGN KEY (ciclist) REFERENCES Ciclist(id_ciclist) ON DELETE CASCADE,  
    FOREIGN KEY (sponsor) REFERENCES Sponsor(id_sponsor) ON DELETE CASCADE  
);
```

```
INSERT INTO Promovari VALUES(480,13,'2022-09-01');  
INSERT INTO Promovari VALUES(490,14,'2022-09-03');  
INSERT INTO Promovari VALUES(490,16,'2022-09-02');  
INSERT INTO Promovari VALUES(500,17,'2022-09-03');  
INSERT INTO Promovari VALUES(500,19,'2022-09-06');  
INSERT INTO Promovari VALUES(520,13,'2022-09-07');  
INSERT INTO Promovari VALUES(530,20,'2022-09-02');  
INSERT INTO Promovari VALUES(530,17,'2022-09-03');  
INSERT INTO Promovari VALUES(540,18,'2022-09-12');  
INSERT INTO Promovari VALUES(540,19,'2022-09-11');  
INSERT INTO Promovari VALUES(540,13,'2022-09-21');
```

```
CREATE TABLE Medici(  
    id_medic INT PRIMARY KEY,  
    nume VARCHAR(50) NOT NULL,  
    varsta INT NOT NULL,  
    exSp_anterioara BIT DEFAULT 0,  
);  
INSERT INTO Medici VALUES(112,'Ion Popescu', 29, 1);  
INSERT INTO Medici VALUES(911,'John Doe', 41, 1);  
INSERT INTO Medici VALUES(113,'Elena Andronie', 32, 0);  
INSERT INTO Medici VALUES(114,'Melinte Mihai', 51, 1);  
INSERT INTO Medici VALUES(912,'John Deer', 54, 0);
```

```

CREATE TABLE Traseu(
  id_traseu INT PRIMARY KEY,
  lungime FLOAT NOT NULL,
  tip CHAR
);

INSERT INTO Traseu VALUES(1, 2.19, 'A');
INSERT INTO Traseu VALUES(2, 3.32, 'B');
INSERT INTO Traseu VALUES(3, 1.74, 'C');
INSERT INTO Traseu VALUES(4, 4.02, 'D');

CREATE TABLE Superpromotie(
  ciclist INT,
  traseu INT,
  timp INT NOT NULL,
  castiga BIT DEFAULT 0
  PRIMARY KEY (ciclist,traseu)
  FOREIGN KEY (ciclist) REFERENCES Ciclist(id_ciclist) ON DELETE CASCADE,
  FOREIGN KEY (traseu) REFERENCES Traseu(id_traseu),
);

```

```

INSERT INTO Superpromotie VALUES(480,1, 367,0);
INSERT INTO Superpromotie VALUES(490,1, 421,0);
INSERT INTO Superpromotie VALUES(520,2, 567,0);
INSERT INTO Superpromotie VALUES(570,2, 556,0);
INSERT INTO Superpromotie VALUES(550,3, 423,0);
INSERT INTO Superpromotie VALUES(500,3, 367,0);
INSERT INTO Superpromotie VALUES(590,3, 381,0);

```

```
-- CORRELATED SQL
```

```

CREATE TABLE Orar(
  grupa INT,
  traseu INT,
  medic INT,
  PRIMARY KEY(grupa, traseu, medic),
  FOREIGN KEY (grupa) REFERENCES Grupa(id_grupa) ON DELETE CASCADE,
  FOREIGN KEY (traseu) REFERENCES Traseu(id_traseu) ON DELETE CASCADE,
  FOREIGN KEY (medic) REFERENCES Medici(id_medic) ON DELETE CASCADE,
);

```

```

INSERT INTO Orar VALUES(11,1,112);
INSERT INTO Orar VALUES(11,3,113);
INSERT INTO Orar VALUES(12,2,912);
INSERT INTO Orar VALUES(12,4,114);
INSERT INTO Orar VALUES(13,1,113);
INSERT INTO Orar VALUES(13,4,911);

```

```

SELECT DISTINCT Ciclist.num AS Nume, Ciclist.prenume AS Prenume FROM Orar
JOIN Medici
ON Orar.medic = Medici.id_medic
JOIN Traseu
ON traseu = Traseu.id_traseu
JOIN Grupa
ON grupa = Grupa.id_grupa
JOIN Alocat
ON grupa = Alocat.grupa_
JOIN Ciclist
ON ciclist_ = id_ciclist
WHERE Medici.num = 'Elena Andronie' AND tip = 'C'
ORDER BY Ciclist.num;

```

```

-- identificati toti ciclistii
-- sponsorizati de firme
-- al caror domeniu este international (.com)
-- UNCORRELATED SQL bullet

SELECT CONCAT(CONCAT(p.num, ' '), p.prenume) AS "Full Name"
FROM Ciclist p
WHERE p.id_ciclist IN (
    SELECT ciclist
    FROM Promovari
    WHERE sponsor IN (
        SELECT id_sponsor
        FROM Sponsor
        WHERE website LIKE '%.com'
    )
);

```

108 %

Results Messages

	Full Name
1	Remy Metailler
2	Vali Holl
3	Remy Metailler
4	Troy Brosnan
5	Jordie Lynn
6	Corinne Sutter

```
-- Determinati numarul de ciclisti
-- al caror numar de telefon incepe cu 9 SAU 2

SELECT COUNT(ume) AS 'Total', SUBSTRING(telefon, 1,2) AS 'REGIUNE'
FROM Ciclist
GROUP BY SUBSTRING(telefon, 1,2)
HAVING SUBSTRING(telefon, 1,2) = '+9' OR SUBSTRING(telefon, 1,2) = '+2';
```

Results Messages

	Total	REGIUNE
1	3	+2
2	2	+9

```
-- Aratati tabelul ciclistilor
-- dar alaturand o coloana care
-- va determina categoria de varsta la care se incadreaza
-- demo NVL, CASE, fct calendaristice, IIF

SELECT
  CONCAT(CONCAT(p.ume, ' '), p.prenume) AS 'Full Name',
  ISNULL (IIF(DATEPART(d, p.data_nastere) = DATEPART(d, GETDATE()) AND
    DATEPART(m, P.data_nastere) = DATEPART(m, GETDATE()),
    'Birthday', NULL), 'Not birthday' )
    AS 'Celebrates Birthay?',
  CASE
    WHEN data_nastere > '2000-01-01' THEN 'Teen'
    WHEN data_nastere > '1985-01-01' THEN 'Adult'
    WHEN DATEPART(d, p.data_nastere) = DATEPART(d, GETDATE()) AND DATEPART(m, P.data_nastere) = DATEPART(m, GETDATE()) THEN 'La multi ani!'
    ELSE 'Senior'
  END AS 'Categorie'
FROM Ciclist p;
```

108 %

Results		Messages	
	Full Name	Celebrates Birthay?	Categorie
1	Remy Metailler	Not birt	Senior
2	Vali Holl	Not birt	Adult
3	Remy Metailler	Not birt	Senior
4	Vali Holl	Not birt	Adult
5	Troy Brosnan	Not birt	Adult
6	Jordie Lynn	Not birt	Adult
7	Corinne Sutter	Not birt	Teen
8	Nicole Schimdhofe	Not birt	Teen
9	Mihaela Shiffrin	Not birt	Adult
10	Brandon Semenuk	Not birt	Teen
11	Brett Rheeder	Not birt	Adult

Query executed successfully.

```
--CLAUZA WITH
-- DETERMINATI TRASEELE UNDE
-- TIMPII DE PARCURS SUNT
-- Mai buni decat media timpilor din superpromotie

WITH totalTime(traseu, timp) AS (
    SELECT traseu, AVG(timp)
    FROM Superpromotie
    GROUP BY traseu
),

promoavg(avgtime) AS (
    SELECT AVG(timp)
    FROM Superpromotie
)
SELECT traseu
FROM totalTime, promoavg
WHERE totalTime.timp < promoavg.avgtime;
```

	traseu
1	1
2	3

```
-- Setati superpromotie
-- ca fiind castigata
-- de catre runner-up-erii ai fiecarui traseu

UPDATE Superpromotie
SET castiga = 1
WHERE timp IN(
    SELECT MIN(timp) AS 'Timp realizat' FROM Superpromotie
    GROUP BY traseu
);
```

```
--Adaugati 19 puncte tuturor
-- ciclistilor care au castigat superpromotia

UPDATE Ciclist
SET punctaj = punctaj + 19
WHERE id_ciclist IN (
    SELECT ciclist
    FROM Superpromotie
    WHERE castiga = 1
);
```

Results		Messages					
	id_ciclist	nume	prenume	sex	data_nastere	telefon	punctaj
1	480	Remy	Metailler	1	1984-07-12	+812382189	19
2	490	Vali	Holl	0	1995-02-12	+1312321431	0
3	500	Remy	Metailler	1	1984-07-12	+812382189	19
4	510	Vali	Holl	0	1995-02-12	+1312321431	0
5	520	Troy	Brosnan	1	1989-01-02	+21321321421	0
6	530	Jordie	Lynn	1	1987-11-11	+213129321	0
7	540	Corinne	Sutter	0	2000-03-07	+312242141	0
8	550	Nicole	Schimdhofer	0	2001-09-11	+421893417	0
9	560	Mihaela	Shiffrin	0	1998-08-27	+4019238129	0
10	570	Brandon	Semenuk	1	2001-01-16	+918213123	19
11	580	Brett	Rheeder	1	1995-12-29	+94738475	0
12	590	T		1	1999-11-01	+330310310	0


```

-- CORRELATED SQL
-- Descoperiti id-urile ciclistilor
-- care au scos timp mai bun in SUPERPROMOTIE
-- fata de media traseului

SELECT DISTINCT nume, prenume
FROM Ciclist
WHERE Ciclist.id_ciclist IN
(
    SELECT p.ciclist
    FROM Superpromotie p
    WHERE timp <
        (
            SELECT AVG(timp)
            FROM Superpromotie q
            WHERE p.traseu = q.traseu
        )
);

```

Results		Messages	
	nume	prenume	
1	Remy	Metallier	
2	Brandon	Semenuk	
3	Tom	van Steenbergen	

Results		Messages	
	grupa	traseu	medic
1	11	1	112
2	11	3	113
3	12	2	912
4	12	4	114
5	13	1	113
6	13	4	911

```

-- Nu mai exista buget pentru medicul 113
-- Afisati noul orar

SELECT * FROM Orar;
DELETE FROM Medici
WHERE id_medic = '113';

```

Results		Messages	
	grupa	traseu	medic
1	11	1	112
2	12	2	912
3	12	4	114
4	13	4	911

```
-- UN VIEW
```

```
CREATE VIEW [Ciclist Punctaj] AS
SELECT nume, prenume, punctaj
FROM Ciclist;
SELECT * FROM [Ciclist Punctaj];
```

Results		Messages	
	nume	prenume	punctaj
1	Remy	Metailler	19
2	Vali	Holl	0
3	Remy	Metailler	19
4	Vali	Holl	0
5	Troy	Brosnan	0
6	Jordie	Lynn	0
7	Corinne	Sutter	0
8	Nicole	Schimdhofer	0
9	Mihaela	Shiffrin	0
10	Brandon	Semenuk	19
11	Brett	Rheeder	0

✓ Query executed successfully.

```
SELECT * FROM ciclist;
SELECT * from Sponsor;
CREATE SEQUENCE SEQ_CICL
START WITH 470
INCREMENT BY 10
MAXVALUE 1350;
```