



Baze de date-anul I



Gestionarea proiectului Smarthack



Nechita Maria-Ilinca

Baze de date- anul I

Seria 14

GESTIUNEA PROIECTULUI SMARTHACK

Nechita Maria-Ilinca

grupa 143

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



- Modelul dat reprezintă o bază de date care cuprinde informații în legătură cu modalitatea de organizare și desfășurare a proiectului Smarthack, un hackathon de 24 de ore în care participanții vor dezvolta o aplicație pe baza unei teme primite de la firmele partenere.
- Acesta este util pentru a reține toate datele legate de organizarea evenimentului, mai exact informații legate de firme, teme, echipe, voluntari și premii.
- În primul rând, la eveniment vor participa mai multe firme din domeniul IT partenere care pot alege un pachet total sau unul partial, care va cuprinde diverse facilități de promovare. Acesta vor putea susține câteva traininguri tematice pentru participanți. Fiind vorba despre un concurs, vor exista și premii date de firmele care au propus tema, dar și premii speciale date de către alte firme partenere.
- Fiecare dintre firme va veni cu câte o temă pe baza căreia participanții să își dezvolte aplicația, iar la fiecare firma pot exista mai multe echipe care să concureze pentru premiul oferit de aceasta. De asemenea, fiecare firma va trimite mentori, care să îndrume participanții pe parcursul competiției.
- Pe lângă firme, vor exista și alți sponsori care să asigure mâncare și băutură pe durata evenimentului. Pentru ca totul să se desfășoare bine, în cadrul acestuia vor exista și voluntari, care să se asigure că participanții nu au probleme pe durata concursului.

2. Restricții de funcționare



Modelul de date respectă următoarele constrângeri:

- **La un eveniment** iau parte mai multe firme. Acesta se desfășoară într-o singură **Locație**.
- La eveniment participă mai mulți **Voluntari** și poate fi sponsorizat de unul sau mai mulți **Sponsori** (pentru băutură, mâncare, alte nevoi).
- Evenimentul va avea parte și de o **Promovare** corespunzătoare pe rețele de socializare, dar realizată și prin mai multe **Materiale** de promovare.
- O **firma** participantă trimite unul sau mai mulți **Mentori** care se ajute echipele. Aceasta poate alege un singur pachet (Gold sau Silver) și poate acorda unul sau mai multe **Premii**. De asemenea, firma alege câteva echipe din cele înscrise ca să concureze pentru aceasta.
- Firmele vor putea ține câte un **training**.
- O **echipa**: se poate înscrie la una sau mai multe firme, primește un **Premiu** sau mai multe și este formată din mai mulți participanți. Premiile pot fi acordate uneia sau mai multor Echipe.

3. Entități



ENTITATE	CHEIE PRIMARĂ	OBSERVAȚII
Eveniment	id_eveniment	Informații generale despre eveniment (data)
Locatie	id_locatie	Informații legate de locație (numărul de camere disponibile, nume, adresă și date de contact)
Voluntar	id_voluntar	Reține numele voluntarului, datele sale de contact și tura la care acesta a luat parte.
Promovare	id_promovare	Cuprinde date legate de promovarea evenimentului, precum data de început a acesteia și rețele de socializare prin intermediul căreia s-a realizat.
Material	id_material	Cum promovarea se va realiza și prin intermediul materialelor promoționale, dorim sa știm tipul acestora și numărul.
Sponsor	id_sponsor	Pentru ceilalți sponsori dorim să știm numele acestora, tipul de produs furnizat și numărul de produse aduse.
TrainingFirma	id_firma	Cuprinde date legate de trainingurile firmelor, precum data acestora și numele.

3. Entități



ENTITATE	CHEIE PRIMARĂ	OBSERVAȚII
PachetFirma	id_pachet	Există 2 tipuri de pachete (Gold și Silver, reprezentate drept subentități), care vor oferi diverse facilități (promovare, mai mulți mentori trimiși de firme) și vor avea prețuri diferite.
Echipa	id echipa	Pentru fiecare echipă reținem numărul de participanți și numele echipei.
Participant	id_participant	Despre fiecare participant vom cunoaște informațiile: numele acestuia, facultatea la care este student, domeniu de studii și anul.
Premiu	id_premiu	Entitatea premiu reține despre fiecare premiu suma de bani oferită și tipul premiului.
Mentor	id_mentor	Pentru fiecare mentor trimis de către firme știm numele și datele de contact.
Firma	id_firma	Despre firme cunoaștem numele, mail-ul, numărul de telefon și numele temei propuse.

4. Relații



Prezentarea relațiilor modelului de date se va face cât mai amplu și clar. Numele alese pentru relațiile acestui model de date sunt sugestive, reflectând conținutul acestora și entitățile pe care le leagă. Pentru fiecare relație se va preciza cardinalitatea maximă și minimă.

- **EVENTIMENT_participă_FIRMA** = relație care leagă entitățile **EVENTIMENT** și **FIRMA**, reflectând legătura dintre acestea (care sunt firmele participante la eveniment). Ea are cardinalitatea minimă 1:1 (la eveniment participă cel puțin o firmă și o firmă trebuie să participe la eveniment) și cardinalitatea maximă 1: M (la eveniment participă mai multe firme, dar firma poate participa la un singur eveniment).
- **EVENTIMENT_este promovat_PROMOVARE** = relație care leagă entitățile **EVENTIMENT** și **PROMOVARE** reflectând legătura dintre acestea (care sunt metodele de promovare a evenimentului). Ea are cardinalitatea 1:1 (evenimentul este promovat printr-un singur proces de promovare și un proces de promovare corespunde unui singur eveniment).
- **EVENTIMENT_participă_VOLUNTAR** = relație care leagă entitățile **EVENTIMENT** și **VOLUNTAR**, reflectând legătura dintre acestea (ce voluntari ajută în organizare). Ea are cardinalitatea minimă 1:1 (la eveniment participă cel puțin un voluntar) și cardinalitatea maximă 1: M (la eveniment participă mai mulți voluntari, dar voluntarii participă la un singur eveniment).
- **EVENTIMENT_este sponsorizat_SPONSOR** = relație care leagă entitățile **EVENTIMENT** și **SPONSOR** reflectând legătura dintre acestea (care sunt sponsorii evenimentului). Ea are cardinalitatea minimă 1:0 (evenimentul poate să nu fie sponsorizat) și maximă de 1:M (pot exista mai mulți sponsori).

4. Relații



- **EVENIMENT_se desfășoară_LOCATIE** = relație care leagă entitățile **EVENIMENT** și **LOCATIE**, reflectând legătura dintre acestea (unde se va ține hackathonul). Ea are cardinalitatea 1:1 (un eveniment se va desfășura într-o singură locație)
- **PROMOVARE_realizată prin_MATERIAL** = relație care leagă entitățile **PROMOVARE** și **MATERIAL**, reflectând legătura dintre acestea (ce materiale sunt folosite în promovarea evenimentului). Ea are cardinalitatea minimă 1:1 (poate fi folosit un singur tip de material) și cardinalitatea maximă 1: M (pentru o singură promovare pot fi folosite mai multe tipuri de materiale).
- **FIRMA_alege_PACHETFIRMA** = relație care leagă entitățile **FIRMA** și **PACHETFIRMA** reflectând legătura dintre acestea (ce pachete sunt alese de firme). Ea are cardinalitatea 1:1 (o firmă alege un singur pachet).
- **FIRMA_trimite_MENTOR** = relație care leagă entitățile **FIRMA** și **MENTOR**, reflectând legătura dintre acestea (care sunt mentorii trimiși de firme). Ea are cardinalitatea minimă 1:1 (o firmă trimite un singur mentor) și cardinalitatea maximă 1 : M (o firmă poate trimite mai mulți mentori).
- **FIRMA_oferă_PREMIU**= relație care leagă entitățile **FIRMA** și **PREMIU**, reflectând legătura dintre acestea (ce premii oferă firmele concurenților). Ea are cardinalitatea minimă 1:1 (o firmă poate acorda un premiu) și cardinalitatea maximă 1 : M (o firmă poate acorda oricâte premii).




4. Relații



- FIRMA_are_ECHIPA = relație care leagă entitățile FIRMA și ECHIPA reflectând legătura dintre acestea (ce echipe se înscriu pentru fiecare firmă în parte). Ea are cardinalitatea minimă 1:1 (o firmă are minim o echipă înscrisă) și cardinalitate maximă M:M (mai multe echipe se pot înscrie pentru a concura la mai multe firme).
- ECHIPA_este_formată_din_PARTICIPANT = relație care leagă entitățile ECHIPA și PARTICIPANT, reflectând legătura dintre acestea (care sunt participanții). Ea are cardinalitatea minimă 1:1 (o echipă are minim un membru) și maximă 1:M (o echipă poate avea oricât de mulți membri).
- ECHIPA_câștigă_PREMIU = relație care leagă entitățile ECHIPA și PREMIU, reflectând legătura dintre acestea (care sunt premiile câștigate de fiecare echipă). Ea are cardinalitatea minimă 1:0 (o echipă poate să nu câștige niciun premiu) și maximă M:M (o echipă poate câștiga mai multe premii și un premiu poate fi câștigat de mai multe echipe- de exemplu: cea mai originală idee, cea mai rapidă aplicație).
- FIRMA_tine_TRAININGFIRMA= relație care leagă entitățile FIRMA și TRAININGFIRMA reflectând legătura dintre acestea (ce traininguri sunt ținute fiecare firmă în parte). Ea are cardinalitatea 1:1 (o firmă ține un training și fiecare training este ținut de o firmă).

5. Attribute








Eveniment			
	id_eveniment	int	
	DataEveniment	date	

Entitatea independentă EVENIMENT are ca attribute:

id_eveniment = variabilă de tip întreg, care reprezintă codul unui eveniment.

DataEveniment = variabilă de tip dată calendaristică, care reprezintă data la care se desfășoară evenimentul.

Locatie			
	id_locatie	int	
	NumeLocatie	varchar(255)	
	NumarCamereDisponibile	int	
	ContactLocatie	varchar(255)	






Entitatea LOCATIE are ca attribute:

id_locatie = variabilă de tip întreg, care reprezintă codul unei locații.

NumeLocatie = variabilă de tip caracter, de lungime maximă 255, care reprezintă numele locației.

NumarCamereDisponibile = variabilă de tip întreg, care indică numărul încăperilor dintr-o locație.

ContactLocație= variabilă de tip caracter, de lungime maximă 255, care reprezintă datele de contact ale locației.

Promovare			
	id_promovare	int	
	id_eveniment	int	
	ReteaSocializare	varchar(255)	
	DataPromovare	datetime	

Entitatea PROMOVARE are ca attribute:

id_promovare = variabilă de tip întreg, care reprezintă codul unei promovări.





id_eveniment = variabilă de tip întreg, care reprezintă codul evenimentului promovat. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul EVENIMENT. (FK)

ReteaSocializare = variabilă de tip caracter, care reprezintă rețeaua de socializare pe care se face promovarea.

DataPromovare = variabilă de tip caracter, care reprezintă data de la care se începe promovarea..

5. Attribute



Sponsor			
	id_sponsor	int	
	NumeSponsor	varchar(255)	
	TipProduce	varchar(255)	
	NumarProduce	int(255)	






Entitatea SPONSOR are ca atribute:

id_sponsor = variabilă de tip întreg, care reprezintă codul unui sponsor.

NumeSponsor= variabilă de tip caracter, de lungime maximă 255, care reprezintă numele sponsorului.

TipProduce=variabilă de tip caracter, de lungime maximă 255, care reprezintă numele produsului primit.

NumarProduce= variabilă de tip întreg, care indică numărul produselor primite.

Voluntar			
	id_voluntar	int	
	NumeVoluntar	varchar(255)	
	TuraVoluntar	int	
	ContactVoluntar	varchar(255)	

Entitatea VOLUNTAR are ca atribute:








id_voluntar= variabilă de tip întreg, care reprezintă codul unui voluntar.

NumeVoluntar= variabilă de tip caracter, de lungime maximă 255, care reprezintă numele voluntarului.

TuraVoluntar = variabilă de tip întreg, care indică tura în care voluntarul se află la eveniment (1 sau 2). ContactVoluntar= variabilă de tip caracter, de lungime maximă 255, care reprezintă datele de contact ale voluntarului.

5. Attribute













Material			
 id_promovare	int		
TipMaterial	varchar(255)		
NumarMaterial	int		

Entitatea MATERIAL are ca atribute:

id_promovare = variabilă de tip întreg, care reprezintă codul promovării evenimentului. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PROMOVARE.(FK)

TipMaterial= variabilă de tip caracter, de lungime maximă 255, care reprezintă tipul materialului de promovare.

NumarMaterial = variabilă de tip întreg, care indică numărul materialelor de un anumit tip.

Firma			
 id_firma	int		
MailFirma	varchar(255)		
NumarTelefonFirma	int		
NumeTemaPropusa	varchar(255)		
NumeFirma	varchar(50)		

Entitatea FIRMA are ca atribute:








id_firma = variabilă de tip întreg, care reprezintă codul unei firme.

NumeFirma = variabilă de tip caracter, de lungime maximă 255, care reprezintă numele firmei.

NumarTelefonFirma = variabilă de tip caracter, de lungime maximă 255, care reprezintă numărul de telefon al firmei.

NumeTemaPropusă= variabilă de tip caracter, de lungime maximă 255, care reprezintă tema propusă de o firma

MailFirma=variabilă de tip caracter, de lungime maximă 255, care reprezintă mail-ul firmei.

Echipa			
 id echipa	int		
Nume echipa	varchar(255)		
NumarParticipanti	int		

Entitatea ECHIPA are ca atribute:





id echipa = variabilă de tip întreg, care reprezintă codul unei echipe.

Nume echipa= variabilă de tip caracter, de lungime maximă 255, care reprezintă numele unei echipe.

NumarParticipanti= variabilă de tip întreg, care indică numărul de persoane într-o echipă..

5. Attribute










Mentor			
	id_mentor	int	
	NumeMentor	varchar(255)	
	ContactMentor	varchar(255)	

Entitatea MENTOR are ca attribute:

id_mentor = variabilă de tip întreg, care reprezintă codul unui mentor.

NumeMentor= variabilă de tip caracter, de lungime maximă 255, care reprezintă numele mentorului.

ContactMentor= variabilă de tip aracter, de lungime maximă 255, care reprezintă datele de contact ale unui mentor.

Participant			
	id echipa	int	
	id_participant	int	
	NumeParticipant	varchar(255)	
	Facultate	varchar(255)	
	Domeniu	varchar(255)	
	AnStudiu	int	

Entitatea PARTICIPANT are ca attribute:

id_participant = variabilă de tip întreg, care reprezintă codul unui concurent.

id echipa = variabilă de tip întreg, care reprezintă codul echipei.

Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ECHIPA(FK)

NumeParticipant= variabilă de tip caracter, de lungime maximă 255, care reprezintă numele participantului.






Facultate= variabilă de tip aracter, de lungime maximă 255, care reprezintă facultatea la care studiază participantul.

Domeniu= variabilă de tip aracter, de lungime maximă 255, care reprezintă domeniul de studii al participantului.

AnStudiu=variabilă de tip întreg, care indică anul de studiu al unui participant.

5. Attribute



Premiu			
	id_premiu	int	
	id_firma	int	
	Tip_premiu	varchar(255)	
	SumaPremiu	varchar(255)	

Entitatea PREMIU are ca attribute:




id_premiu = variabilă de tip întreg, care reprezintă codul unui premiu.

id_firma = variabilă de tip întreg, care reprezintă codul firmei.

Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FIRMA.(FK)

Tip_premiu= variabilă de tip caracter, de lungime maximă 255, care reprezintă tipul premiului.

SumaPremiu=variabilă de tip întreg, care reprezintă suma de bani oferită ca premiu.

PachetFirma			
	id_firma	int	
	id_pachet	int	

Entitatea PACHETFIRMA are ca attribute:

id_pachet = variabilă de tip întreg, care reprezintă codul unui pachet. .

id_firma = variabilă de tip întreg, care reprezintă codul firmei.

Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FIRMA.(FK)

Subentitatea GOLD are ca attribute:

id_pachet = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui pachet.

SumaPachet= variabilă de tip numeric (real)

FacilitatiPachet=variabilă de tip caracter, indică facilitățile

Subentitatea SILVER are ca attribute:

id_pachet = variabilă de tip întreg, de lungime maximă 5, care reprezintă codul unui pachet.

SumaPachet= variabilă de tip numeric (real)

FacilitatiPachet=variabilă de tip caracter, indică facilitățile





Entitatea TRAININGFIRMA are ca attribute:

id_firma = variabilă de tip întreg, care reprezintă codul firmei.

Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FIRMA.(FK)

NumeTraining=variabilă de tip caracter, de lungime maximă 255, care reprezintă numele trainingului.

DataTraining=variabilă de tip data, care reprezintă data la care se tine trainingul.

TrainingFirma			
	id_firma	int	
	NumeTraining	varchar(255)	
	DataTraining	date(255)	

5. Attribute



Relația FIRMA_are_ECHIPA are ca attribute:

id_firma = variabilă de tip întreg, care reprezintă codul unei firme. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul FIRMA.

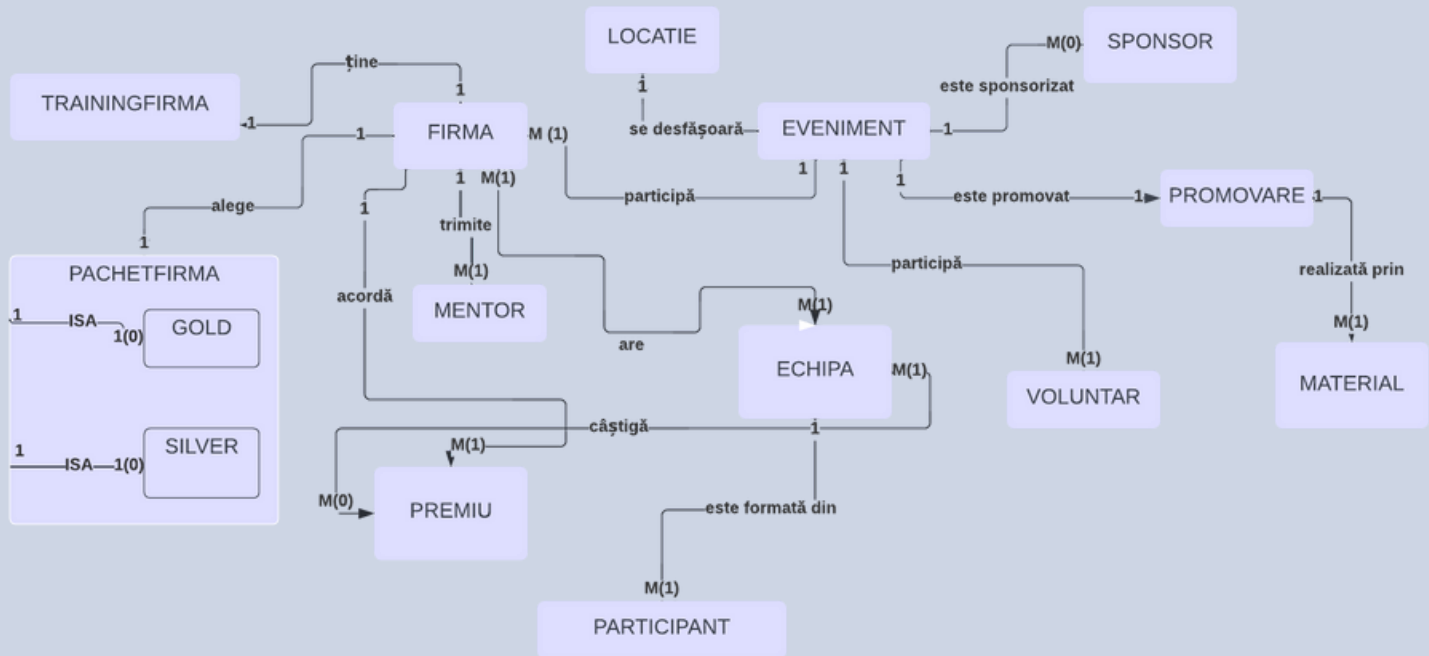
id echipa = variabilă de tip întreg, care reprezintă codul unei echipe. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ECHIPA.

Relația ECHIPA_câștigă_PREMIU are ca attribute:

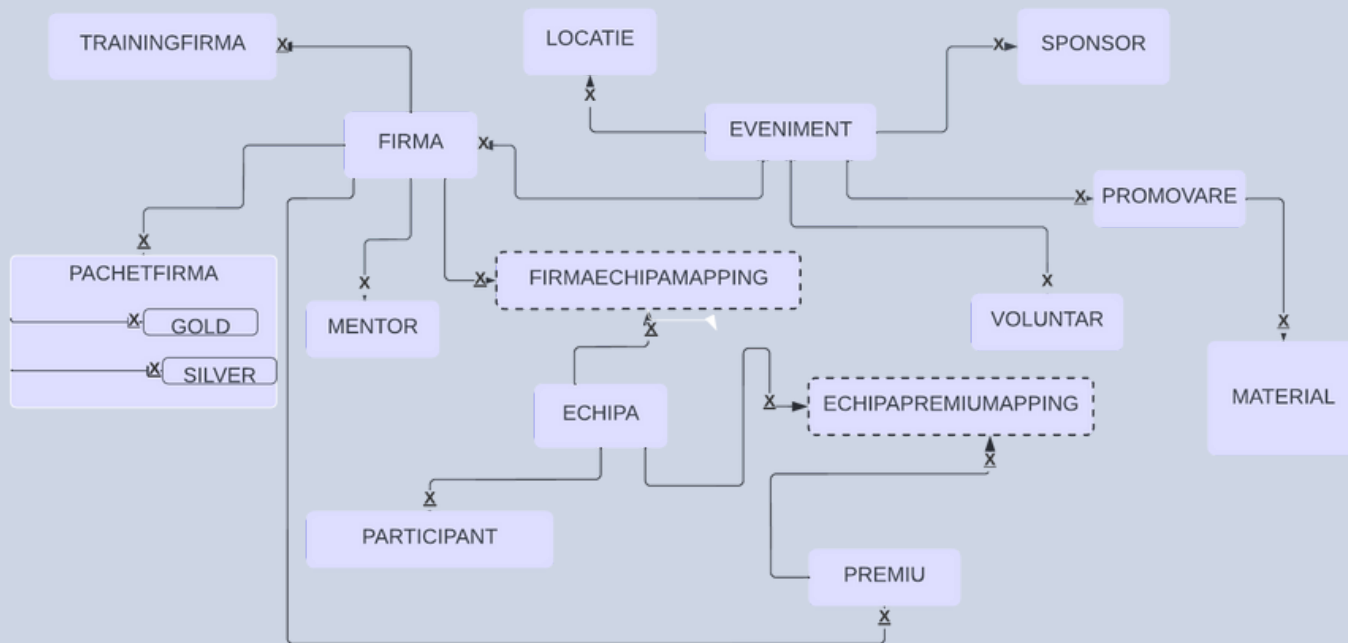
id echipa = variabilă de tip întreg, care reprezintă codul unei echipe. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul ECHIPA.

id_premiu = variabilă de tip întreg, care reprezintă codul unui premiu. Atributul trebuie să corespundă la o valoare a cheii primare din tabelul PREMIU.

6. Diagrama entitate-relație



7. Diagrama conceptuală



8. Scheme relaționale care corespund diagramei conceptuale



- EVENIMENT (id_eveniment#, DataEveniment)
- LOCATIE(id_locatie#, NumeLocatie, NumarCamereDisponibile, ContactLocatie)
- PROMOVARE(id_promovare#, ReteaSocializare, DataPromovare)
- SPONSOR(id_sponsor#, id_eveniment#, NumeSponsor, TipProduce, NumarProduce)
- VOLUNTAR(id_voluntar#, NumeVoluntar, TuraVoluntar, ContactVoluntar)
- MATERIAL (id_promovare#, TipMaterial, NumarMaterial)

- FIRMA(id_firma#, MailFirma, NumarTelefonFirma, NumeTemaPropusa, NumeFirma)
- TRAININGFIRMA(id_firma#, NumeTraining, DataTraining)
- ECHIPA(id echipa#, Nume echipa, NumarParticipanti)
- MENTOR(id_mentor#, NumeMentor, ContactMentor)
- PACHETFIRMA(id_firma#, id_pachet)
- PREMIU(id_premiu#, id_firma#, Tip_premiu, SumaPremiu)
- PARTICIPANT(id echipa#, id_participant#, NumeParticipant, Facultate, Domeniu, AnStudiu)

- ARE(id_firma#, id echipa#)
- CÂȘTIGĂ(id echipa#, id_premiu#)

9. Normalizare: FN1-FN3



Forma normală 1 (FN1)

- O relație se află în prima formă normală dacă fiecărui atribut care o compune îi corespunde o valoare indivizibilă. Mai precis, o relație se află în forma normală 1 dacă există un identificator unic.
- Pentru a demonstra că relațiile avute în diagramă sunt în forma normală 1 se vor lua câteva exemple. De exemplu, în entitatea FIRMA cheia primară este `id_firma`, ceea ce înseamnă că este un identificator unic (fiecare firmă are un id unic). În acest fel, atributul `id_firma` fiind o valoare indivizibilă, relația se află în forma normală 1 (există identificator unic pentru toate intrările din tabel).

Pentru a exemplifica cum se aplică această normalizare se va crea un exemplu imaginar, mai precis se va presupune că pentru o competiție există mai multe tipuri de taxe de participare (`taxă_participare`). Cum arată inițial relația SPONSOR nonFN1:

<code>id_sponsor</code>	<code>NumeSponsor</code>	<code>TipProduse</code>	<code>NumarProduse</code>
1	TrentaPizza	pizza, suc	100,200
2	Red Bull	energizant	200

Astfel, se poate remarca că relația mai sus menționată nu este în FN1, deoarece atributelor `TipProduse` și `NumarProduse` nu le corespund valori indivizibile.

9. Normalizare: FN1-FN3



- După ce a fost aplicată prima formă normală, relația va arăta astfel:

id_sponsor	NumeSponsor	TipProduse	NumarProduse
1	TrentaPizza	pizza	100
1	TrentaPizza	suc	200
2	Red Bull	energizant	200

- Se poate observa că după normalizare fiecărui atribut care o compune îi corespunde o valoare indivizibilă

Forma normală 2 (FN2)

- Definiția acestei forme normale este următoarea. O relație se află în forma normală 2 dacă și numai dacă aceasta se află în prima formă normală și dacă fiecare atribut care nu face parte din cheie este dependent de întreaga cheie primară.
- Pentru a exemplifica această formă normală se va arăta un exemplu care pornește de la o relație care inițial nu era în forma normală 2 și care se va transforma într-o relație din FN2.

9. Normalizare: FN1-FN3



Inițial relația de la PROMOVARE conținea și atributele TipMaterial și NumarMaterial. Aceasta se află în FN1, deoarece există identificator unic pentru toate intrările din tabel.

- Pentru a fi în FN2 trebuie ca fiecare atribut care nu este cheie să fie dependent de întreaga cheie primară. În acest caz, atributele TipMaterial și NumarMaterial trebuie să depindă de întreaga cheie primară id_promovare# și id_eveniment#, dar se poate observa faptul că acestea nu depind direct de toată cheia primară.
- Acest fapt se explică prin existența dependenței directe dintre id_promovare#, TipMaterial și NumarMaterial, ceea ce înseamnă că , TipMaterial și NumarMaterial doar de o parte a cheii primare, anume de id_promovare#. Astfel, relația prezentată mai sus nu se află în a doua formă normală.
- Astfel, dependențele vor fi:
 - o {id_promovare#}-> {TipMaterial, NumarMaterial}
 - o {id_rezultat#, id_eveniment#}->{ReteaSocializare, DataPromovare}
- Transformarea în FN2 se face alcătuind încă un tabel: Material, în care vom avea id_promovare și cele două atribute menționate anterior.

9. Normalizare: FN1-FN3



Forma normală 3 (FN3)

- Definiția acestei forme normale este următoarea. O relație se află în FN3 dacă și numai dacă se află în FN2 și dacă fiecare atribut care nu este cheie, depinde direct de cheia primară și numai de ea.
- Pentru exemplificarea acestei forme normale am ales să se realizeze un exemplu fictiv pentru PREMIU (se presupune că atributul SumaPremiu depinde de Tip_premiu, fapt ce nu este adevărat în cazul diagramei construite). Astfel, dacă ar fi aceste date în PREMIU (SumaPremiu de la id_premiu este creată doar pentru exemplu, valoarea reală inserată fiind alta).

id_premiu	id_firma	Tip_premiu	SumaPremiu
1	1	Innovation	200
2	2	Premium	500
3	3	BestApp	1000

- Această relație este în prima formă normală, pentru că există identificator unic pentru toate intrările din tabel, anume id_premiu#.
- Se află și în FN2, deoarece se poate remarca faptul că attributele care nu sunt în componența cheii primare sunt dependent de întreaga cheie primară.

9. Normalizare: FN1-FN3



- În acest exemplu se observă că atributul SumaPremiu depinde tranzitiv de cheia primară id_premiu# prin intermediul atributului Tip_premiu.
- Astfel, acest exemplu nu se află în FN3, având în vedere relațiile de dependență întâlnite:
 - o {id_premiu#,id_firma#} -> {Tip_premiu}
 - o {id_premiu#,id_firma#}-> {Tip_premiu}->{SumaPremiu}
- Pentru a aduce relația antrenor_nonFN3 în forma normală 3 se aplică regula Casey-Delobel care presupune descompunerea relației prin eliminarea dependenței funcționale tranzitive. În acest mod, din cauza dependenței funcționale Tip_premiu->SumaPremiu care surprinde faptul că relația nu se află în FN3, se înlocuiește relația prin două proiecții:

o PREMIU_FN3(id_premiu#, Tip_premiu)

o Tip_premiu_SumaPremiu(Tip_premiu, SumaPremiu)

id_premiu	id_firma	Tip_premiu
1	1	Innovation
2	2	Premium
3	3	BestApp

Tip_premiu	SumaPremiu
Innovation	200
Premium	500
BestApp	1000

10. Crearea tabelelor în SQL și inserarea de date în acestea



EVENTIMENT

```
CREATE TABLE [EVENTIMENT] (  
    id_eventiment int CONSTRAINT PKEY_EVENTIMENT PRIMARY KEY,  
    dataeveniment DATE CONSTRAINT dataeveniment_const NOT NULL,  
);  
  
INSERT INTO EVENTIMENT  
VALUES(1, convert(DATETIME, '5-11-2019',103));  
  
INSERT INTO EVENTIMENT  
VALUES(2,convert(DATETIME, '4-11-2020', 103));  
  
INSERT INTO EVENTIMENT  
VALUES(3,convert(DATETIME, '30-10-2021', 103));  
  
INSERT INTO EVENTIMENT  
VALUES(4,convert(DATETIME,'5-11-2022', 103));  
  
INSERT INTO EVENTIMENT  
VALUES(5,convert(DATETIME, '14-07-2010', 103));  
  
SELECT *  
FROM [EVENTIMENT];
```

%

Results Messages

id_eventiment	dataeveniment
1	2019-11-05
2	2020-11-04
3	2021-10-30
4	2022-11-05
5	2010-07-14

```
CREATE TABLE [EVENTIMENT] (  
    id_eventiment int CONSTRAINT PKEY_EVENTIMENT PRIMARY KEY,  
    dataeveniment DATE CONSTRAINT dataeveniment_const NOT NULL,  
);
```

```
INSERT INTO EVENTIMENT  
VALUES(1, convert(DATETIME, '5-11-2019',103));
```

```
INSERT INTO EVENTIMENT  
VALUES(2,convert(DATETIME, '4-11-2020', 103));
```

```
INSERT INTO EVENTIMENT  
VALUES(3,convert(DATETIME, '30-10-2021', 103));
```

```
INSERT INTO EVENTIMENT  
VALUES(4,convert(DATETIME,'5-11-2022', 103));
```

```
INSERT INTO EVENTIMENT  
VALUES(5,convert(DATETIME, '14-07-2010', 103));  
SELECT *FROM [EVENTIMENT]
```


FIRMA

```
CREATE TABLE [FIRMA] (  
  id_firma int IDENTITY(1,1) NOT NULL,  
  NumeFirma varchar(255) NOT NULL,  
  MailFirma varchar(255),  
  NumarTelefonFirma int NOT NULL,  
  NumeTemaPropusa varchar(255) NOT NULL,  
  CONSTRAINT [PK_FIRMA] PRIMARY KEY CLUSTERED  
  (  
    [id_firma] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)
```

```
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Softbinator','contact@softbinator.com','0757404781','AI in Health')  
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Bitdefender','office@bitdefender','0212063470','Application simulating an Operating System')  
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Adobe','NULL','0314133776','Mini-Creative Cloud Application')  
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Luxoft','office@luxoft','0212036600','Capture the flag')  
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Accenture','NULL','1571434500','The art of AI-application')
```

```
SELECT *  
FROM [FIRMA];
```

results Messages

id_firma	NumeFirma	MailFirma	NumarTelefonFirma	NumeTemaPropusa
1	Softbinator	contact@softbinator.com	757404781	AI in Health
2	Bitdefender	office@bitdefender	212063470	Application simulating an Operating System
3	Adobe	NULL	314133776	Mini-Creative Cloud Application
4	Luxoft	office@luxoft	212036600	Capture the flag
5	Accenture	NULL	1571434500	The art of AI-application

```
CREATE TABLE [FIRMA] (  
  id_firma int IDENTITY(1,1) NOT NULL,  
  NumeFirma varchar(255) NOT NULL,  
  MailFirma varchar(255),  
  NumarTelefonFirma int NOT NULL,  
  NumeTemaPropusa varchar(255) NOT NULL,  
  CONSTRAINT [PK_FIRMA] PRIMARY KEY CLUSTERED  
  (  
    [id_firma] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)  
)
```

```
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Softbinator','contact@softbinator.com','0757404781','AI in Health')  
INSERT INTO Firma( NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Bitdefender','office@bitdefender','0212063470','Application simulating an Operating System')  
INSERT INTO Firma(NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Adobe','NULL','0314133776','Mini-Creative Cloud Application')  
INSERT INTO Firma( NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Luxoft','office@luxoft','0212036600','Capture the flag')  
INSERT INTO Firma( NumeFirma, MailFirma, NumarTelefonFirma, NumeTemaPropusa)  
VALUES ('Accenture','NULL','1571434500','The art of AI-application')  
SELECT *  
FROM [FIRMA];
```

LOCATIE



```
CREATE TABLE [LOCATIE] (
    id_locatie int NOT NULL,
    NumeLocatie varchar(255) NOT NULL,
    ContactLocatie varchar(255) NOT NULL,
    NumarCamereDisponibile int NOT NULL,
    CONSTRAINT [PK_LOCATIE] PRIMARY KEY CLUSTERED
    (
        [id_locatie] ASC
    ) WITH (IGNORE_DUP_KEY = OFF)
)

INSERT INTO Locatie(id_locatie,NumeLocatie,ContactLocatie, NumarCamereDisponibile)
VALUES (10,'Impact Hub Universitate','+40 372 893 889',5)
INSERT INTO Locatie( id_locatie,NumeLocatie,ContactLocatie, NumarCamereDisponibile)
VALUES (11,'Impact Hub Floreasca','+40 372 893 889',6)
INSERT INTO Locatie(id_locatie,NumeLocatie,ContactLocatie, NumarCamereDisponibile)
VALUES (12,'Romexpo','+40 21 202 5700',6)
INSERT INTO Locatie( id_locatie,NumeLocatie,ContactLocatie, NumarCamereDisponibile)
VALUES (13,'aSpace Baneasa', 'office@aspace.ro', 6)
INSERT INTO Locatie( id_locatie,NumeLocatie,ContactLocatie, NumarCamereDisponibile)
VALUES (14,'aSpace Floreasca', 'office@aspace.ro', 6)

SELECT *
FROM [LOCATIE];
```

id_locatie	NumeLocatie	ContactLocatie	NumarCamereDisponibile
10	Impact Hub Universitate	+40 372 893 889	5
11	Impact Hub Floreasca	+40 372 893 889	6
12	Romexpo	+40 21 202 5700	6
13	aSpace Baneasa	office@aspace.ro	6
14	aSpace Floreasca	office@aspace.ro	6

```
CREATE TABLE [LOCATIE] (
    id_locatie int NOT NULL,
    NumeLocatie varchar(255) NOT NULL,
    ContactLocatie varchar(255) NOT NULL,
    NumarCamereDisponibile int NOT NULL,
    CONSTRAINT [PK_LOCATIE] PRIMARY KEY CLUSTERED
    (
        [id_locatie] ASC
    ) WITH (IGNORE_DUP_KEY = OFF)
)
```

```
INSERT INTO Locatie(id_locatie,NumeLocatie,ContactLocatie,
NumarCamereDisponibile)
VALUES (10,'Impact Hub Universitate','+40 372 893 889',5)
INSERT INTO Locatie( id_locatie,NumeLocatie,ContactLocatie,
NumarCamereDisponibile)
VALUES (11,'Impact Hub Floreasca','+40 372 893 889',6)
INSERT INTO Locatie(id_locatie,NumeLocatie,ContactLocatie,
NumarCamereDisponibile)
VALUES (12,'Romexpo','+40 21 202 5700',6)
INSERT INTO Locatie( id_locatie,NumeLocatie,ContactLocatie,
NumarCamereDisponibile)
VALUES (13,'aSpace Baneasa', 'office@aspace.ro', 6)
INSERT INTO Locatie( id_locatie,NumeLocatie,ContactLocatie,
NumarCamereDisponibile)
VALUES (14,'aSpace Floreasca', 'office@aspace.ro', 6)
```

```
SELECT *
FROM [LOCATIE];
```

PROMOVARE

```
CREATE TABLE [PROMOVARE] (  
  id_promovare int NOT NULL,  
  id_eveniment int FOREIGN KEY REFERENCES Eveniment(id),  
  ReteaSocializare varchar(255) NOT NULL,  
  DataPromovare datetime NOT NULL,  
  CONSTRAINT [PK_PROMOVARE] PRIMARY KEY CLUSTERED  
  (  
    [id_promovare] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)  
)
```

```
INSERT INTO Promovare(id_promovare, id_eveniment,  
ReteaSocializare,DataPromovare)  
VALUES (1,1,'Facebook',convert(DATETIME, '14-09-2022', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment,  
ReteaSocializare,DataPromovare)  
VALUES (2,1,'Instagram',convert(DATETIME, '15-09-2022', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment,  
ReteaSocializare,DataPromovare)  
VALUES (3,1,'TikTok',convert(DATETIME, '20-09-2022', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment,  
ReteaSocializare,DataPromovare)  
VALUES (4,1,'Site-ASMI',convert(DATETIME, '26-09-2022', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment,  
ReteaSocializare,DataPromovare)  
VALUES (5,1,'Site-facultate',convert(DATETIME, '27-09-2022', 103))
```

```
SELECT *  
FROM [PROMOVARE];
```

```
CREATE TABLE [PROMOVARE] (  
  id_promovare int NOT NULL,  
  id_eveniment int FOREIGN KEY REFERENCES Eveniment(id_eveniment),  
  ReteaSocializare varchar(255) NOT NULL,  
  DataPromovare datetime NOT NULL,  
  CONSTRAINT [PK_PROMOVARE] PRIMARY KEY CLUSTERED  
  (  
    [id_promovare] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)  
)  
  
INSERT INTO Promovare(id_promovare, id_eveniment, ReteaSocializare,DataPromovare)  
VALUES (1,4,'Facebook',convert(DATETIME, '14-09-2022', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment, ReteaSocializare,DataPromovare)  
VALUES (2,4,'Instagram',convert(DATETIME, '15-09-2022', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment, ReteaSocializare,DataPromovare)  
VALUES (3,2,'TikTok',convert(DATETIME, '20-09-2020', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment, ReteaSocializare,DataPromovare)  
VALUES (4,3,'Site-ASMI',convert(DATETIME, '26-09-2021', 103))  
INSERT INTO Promovare(id_promovare, id_eveniment, ReteaSocializare,DataPromovare)  
VALUES (5,1,'Site-facultate',convert(DATETIME, '27-09-2019', 103))  
  
SELECT *  
FROM [PROMOVARE];
```

Results Messages				
id_promovare	id_eveniment	ReteaSocializare	DataPromovare	
1	4	Facebook	2022-09-14 00:00:00.000	
2	4	Instagram	2022-09-15 00:00:00.000	
3	2	TikTok	2020-09-20 00:00:00.000	
4	3	Site-ASMI	2021-09-26 00:00:00.000	
5	1	Site-facultate	2019-09-27 00:00:00.000	

VOLUNTAR

```
CREATE TABLE [VOLUNTAR] (
  id_voluntar int IDENTITY(1,2) NOT NULL,
  ContactVoluntar varchar(255) NOT NULL,
  NumeVoluntar varchar(255) NOT NULL,
  TuraVoluntar varchar(255) NOT NULL,
  CONSTRAINT [PK_VOLUNTAR] PRIMARY KEY CLUSTERED
(
  [id_voluntar] ASC
) WITH (IGNORE_DUP_KEY = OFF)
)

INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('ionel.popescu@asmi.ro', 'Ionel Popescu', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('maria.ionescu@asmi.ro', 'Maria Ionescu', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('andreea.gheorghe@asmi.ro', 'Andreea Gheorghe', 2)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('teodora.marcu@asmi.ro', 'Teodora Marcu', 2)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('luca.andrei@asmi.ro', 'Luca Andrei', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('mihai.alexandru@asmi.ro', 'Mihai Alexandru', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('anca.marinescu@asmi.ro', 'Anca Marinescu', 2)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('mihai.constantin@asmi.ro', 'Mihai Constantin', 2)

SELECT *
FROM [VOLUNTAR];
```

%

Results Messages

id_voluntar	ContactVoluntar	NumeVoluntar	TuraVoluntar
1	ionel.popescu@asmi.ro	Ionel Popescu	1
3	maria.ionescu@asmi.ro	Maria Ionescu	1
5	andreea.gheorghe@asmi.ro	Andreea Gheorghe	2
7	teodora.marcu@asmi.ro	Teodora Marcu	2
9	luca.andrei@asmi.ro	Luca Andrei	1
11	mihai.alexandru@asmi.ro	Mihai Alexandru	1
13	anca.marinescu@asmi.ro	Anca Marinescu	2

```
CREATE TABLE [VOLUNTAR] (
  id_voluntar int IDENTITY(1,2) NOT NULL,
  ContactVoluntar varchar(255) NOT NULL,
  NumeVoluntar varchar(255) NOT NULL,
  TuraVoluntar varchar(255) NOT NULL,
  CONSTRAINT [PK_VOLUNTAR] PRIMARY KEY CLUSTERED
(
  [id_voluntar] ASC
) WITH (IGNORE_DUP_KEY = OFF)
```


```
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('ionel.popescu@asmi.ro', 'Ionel Popescu', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('maria.ionescu@asmi.ro', 'Maria Ionescu', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('andreea.gheorghe@asmi.ro', 'Andreea Gheorghe', 2)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('teodora.marcu@asmi.ro', 'Teodora Marcu', 2)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('luca.andrei@asmi.ro', 'Luca Andrei', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('mihai.alexandru@asmi.ro', 'Mihai Alexandru', 1)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('anca.marinescu@asmi.ro', 'Anca Marinescu', 2)
INSERT INTO Voluntar(ContactVoluntar,NumeVoluntar,TuraVoluntar)
VALUES ('mihai.constantin@asmi.ro', 'Mihai Constantin', 2)
```

```
SELECT *
FROM [VOLUNTAR];
```

MENTOR

```
CREATE TABLE [Mentor] (  
  id_mentor int NOT NULL,  
  NumeMentor varchar(255) NOT NULL,  
  ContactMentor varchar(255) NOT NULL,  
  CONSTRAINT [PK_MENTOR] PRIMARY KEY CLUSTERED  
  (  
    [id_mentor] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF))  
  
CREATE SEQUENCE sequence_2  
start with 1  
increment by 3  
minvalue 0  
maxvalue 100  
cycle;
```

```
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_2 AS int),'Daniel  
Ilinca','https://ro.linkedin.com/in/danielilinc')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_2 AS int),'Rares  
Carligeanu','https://ro.linkedin.com/in/rares-cirligeanu-b34471b7')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_2 AS int),'Florin  
Talpes','https://ro.linkedin.com/in/florin-talpes-a2508a4')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_2 AS int),'Anca Marcu ','https://ro.linkedin.com/in/anca  
marcu-b9693540')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_2 AS int),'Catalin Cosoi','https://ro.linkedin.com › in ›  
catalincosoi')  
SELECT *  
FROM [MENTOR];
```



```
CREATE TABLE [MENTOR] (  
  id_mentor int NOT NULL,  
  NumeMentor varchar(255) NOT NULL,  
  ContactMentor varchar(255) NOT NULL,  
  CONSTRAINT [PK_MENTOR] PRIMARY KEY CLUSTERED  
  (  
    [id_mentor] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)  
)  
  
CREATE SEQUENCE sequence_3  
start with 1  
increment by 3  
minvalue 0  
maxvalue 100  
cycle;  
  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_3 AS int),'Daniel Ilinca','https://ro.linkedin.com/in/danielilinc')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_3 AS int),'Rares Carligeanu','https://ro.linkedin.com/in/rares-cirligeanu-b34471b7')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_3 AS int),'Florin Talpes','https://ro.linkedin.com/in/florin-talpes-a2508a4')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_3 AS int),'Anca Marcu ','https://ro.linkedin.com/in/anca-marcu-b9693540')  
INSERT INTO Mentor VALUES(CAST(NEXT VALUE FOR sequence_3 AS int),'Catalin Cosoi','https://ro.linkedin.com › in › catalincosoi')  
SELECT *  
FROM [MENTOR];
```

id_mentor	NumeMentor	ContactMentor
1	Daniel Ilinca	https://ro.linkedin.com/in/danielilinc
4	Rares Carligeanu	https://ro.linkedin.com/in/rares-cirligeanu-b344...
7	Florin Talpes	https://ro.linkedin.com/in/florin-talpes-a2508a4
10	Anca Marcu	https://ro.linkedin.com/in/anca-marcu-b9693540
13	Catalin Cosoi	https://ro.linkedin.com › in › catalincosoi

SPONSOR

```
CREATE TABLE [SPONSOR] (  
  id_sponsor int NOT NULL,  
  NumeSponsor varchar(255) NOT NULL,  
  TipProduce varchar(255) NOT NULL,  
  NumarProduce int NOT NULL,  
  CONSTRAINT [PK_SPONSOR] PRIMARY KEY CLUSTERED  
  (  
    [id_sponsor] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)  
)
```

```
CREATE SEQUENCE sequence_1  
start with 2  
increment by 2  
minvalue 0  
maxvalue 100  
cycle;
```

```
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'TrentaPizza', 'Pizza', 100);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Sloop', 'Suc fresh', 200);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Red Bull', 'Energizant', 200);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Fru Fru', 'Salata', 100);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Chio', 'Snacksuri', 150);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Aqua Carpatica', 'Apa', 400);
```

```
SELECT *  
FROM [SPONSOR];
```

```
CREATE TABLE [SPONSOR] (  
  id_sponsor int NOT NULL,  
  NumeSponsor varchar(255) NOT NULL,  
  TipProduce varchar(255) NOT NULL,  
  NumarProduce int NOT NULL,  
  CONSTRAINT [PK_SPONSOR] PRIMARY KEY CLUSTERED  
  (  
    [id_sponsor] ASC  
  ) WITH (IGNORE_DUP_KEY = OFF)  
)  
  
CREATE SEQUENCE sequence_1  
start with 2  
increment by 2  
minvalue 0  
maxvalue 100  
cycle;  
  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'TrentaPizza', 'Pizza', 100);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Sloop', 'Suc fresh', 200);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Red Bull', 'Energizant', 200);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Fru Fru', 'Salata', 100);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Chio', 'Snacksuri', 150);  
INSERT INTO Sponsor VALUES (CAST(NEXT VALUE FOR sequence_1 AS int), 'Aqua Carpatica', 'Apa', 400);
```

```
SELECT *  
FROM [SPONSOR];
```

Results Messages			
id_sponsor	NumeSponsor	TipProduce	NumarProduce
2	TrentaPizza	Pizza	100
4	Sloop	Suc fresh	200
6	Red Bull	Energizant	200
8	Fru Fru	Salata	100
10	Chio	Snacksuri	150
12	Aqua Carpatica	Apa	400

MATERIAL

```
CREATE TABLE [MATERIAL] (  
    id_promovare int FOREIGN KEY REFERENCES Promovare(id_promovare),  
    TipMaterial varchar(255) NOT NULL,  
    NumarMaterial int NOT NULL  
    CONSTRAINT [PK_MATERIAL] PRIMARY KEY CLUSTERED  
(  
    [id_promovare] ASC  
) WITH (IGNORE_DUP_KEY = OFF)  
)  
INSERT INTO Material VALUES(1, 'Afise', 10)  
INSERT INTO Material VALUES(2, 'Flyere',100)  
INSERT INTO Material VALUES(3, 'Pixuri', 100)  
INSERT INTO Material VALUES (4, 'Tricouri',150)  
INSERT INTO Material VALUES(5, 'Cani',50)
```

```
select *  
from MATERIAL;
```

Results Messages

id_promovare	TipMaterial	NumarMaterial
1	Afise	10
2	Flyere	100
3	Pixuri	100
4	Tricouri	150
5	Cani	50

```
CREATE TABLE [MATERIAL] (  
    id_promovare int FOREIGN KEY REFERENCES Promovare(id_promovare),  
    TipMaterial varchar(255) NOT NULL,  
    NumarMaterial int NOT NULL  
    CONSTRAINT [PK_MATERIAL] PRIMARY KEY CLUSTERED  
(  
    [id_promovare] ASC  
) WITH (IGNORE_DUP_KEY = OFF)  
)  
INSERT INTO Material VALUES(1, 'Afise', 10)  
INSERT INTO Material VALUES(2, 'Flyere',100)  
INSERT INTO Material VALUES(3, 'Pixuri', 100)  
INSERT INTO Material VALUES (4, 'Tricouri',150)  
INSERT INTO Material VALUES(5, 'Cani',50)
```

```
select *  
from MATERIAL;
```

ECHIPA

```
CREATE TABLE [ECHIPA] (  
    id_echipa int IDENTITY(1,1) NOT NULL,  
    Nume_echipa varchar(255) NOT NULL,  
    NumarParticipanti int NOT NULL  
  
    CONSTRAINT [PK_ECHIPA] PRIMARY KEY CLUSTERED  
    (  
        [id_echipa] ASC  
    ) WITH (IGNORE_DUP_KEY = OFF)  
  
    )  
  
INSERT INTO Echipa VALUES ('console.log(Win)',3)  
INSERT INTO Echipa VALUES ('NullTeam',4)  
INSERT INTO Echipa VALUES ('hackU',3)  
INSERT INTO Echipa VALUES ('FANTASTIC 4',4)  
INSERT INTO Echipa VALUES ('Computer Mechanics',3)  
INSERT INTO Echipa VALUES ('Spark Hex',4)
```

```
select *  
from ECHIPA;
```

```
CREATE TABLE [ECHIPA] (  
    id_echipa int IDENTITY(1,1) NOT NULL,  
    Nume_echipa varchar(255) NOT NULL,  
    NumarParticipanti int NOT NULL  
  
    CONSTRAINT [PK_ECHIPA] PRIMARY KEY CLUSTERED  
    (  
        [id_echipa] ASC  
    ) WITH (IGNORE_DUP_KEY = OFF)  
  
    )  
INSERT INTO Echipa VALUES ('console.log(Win)',3)  
INSERT INTO Echipa VALUES ('NullTeam',4)  
INSERT INTO Echipa VALUES ('hackU',3)  
INSERT INTO Echipa VALUES ('FANTASTIC 4',4)  
INSERT INTO Echipa VALUES ('Computer Mechanics',3)  
INSERT INTO Echipa VALUES ('Spark Hex',4)
```

```
select *  
from ECHIPA;
```

6

Results Messages

id_echipa	Nume_echipa	NumarParticipanti
1	console.log(Win)	3
2	NullTeam	4
3	hackU	3
4	FANTASTIC 4	4
5	Computer Mechanics	3
6	Spark Hex	4

PARTICIPANT

```
CREATE TABLE [PARTICIPANT] (
    id echipa int FOREIGN KEY REFERENCES Echipa(id echipa),
    id_participant int NOT NULL,
    NumeParticipant varchar(255) NOT NULL,
    Facultate varchar(255) NOT NULL,
    Domeniu varchar(255) NOT NULL,
    AnStudiu int NOT NULL,
    CONSTRAINT [PK_PARTICIPANT] PRIMARY KEY CLUSTERED
    (
        [id_participant] ASC
    ) WITH (IGNORE_DUP_KEY = OFF)
)

INSERT INTO Participant VALUES(1,1,'Anghel Maria','Facultatea de Matematica si Informatica','Informatica',1)
INSERT INTO Participant VALUES(1,2,'Popescu Ana','Facultatea de Matematica si Informatica','CTI',1)
INSERT INTO Participant VALUES(2,3,'Andrei Cristiana','Facultatea de Matematica si Informatica','CTI',2)
INSERT INTO Participant VALUES(2,4,'Luca Octavian','Facultatea de Automatica si Calculatoare','CTI',2)
INSERT INTO Participant VALUES(3,5,'Mihai Florin','Facultatea de Automatica si Calculatoare','IS',1)
INSERT INTO Participant VALUES(3,6,'Preda Ioana','Facultatea de Matematica si Informatica','Infomatica',1)
INSERT INTO Participant VALUES(4,7,'Sasu Mihai','Facultatea de Automatica si Calculatoare', 'CTI',1)
INSERT INTO Participant VALUES(4,8,'Ionescu Alin','Facultatea de Matematica si Informatica','Informatica',2)

select *
from PARTICIPANT;
```

id echipa	id_participant	NumeParticipant	Facultate	Domeniu	AnStudiu
1	1	Anghel Maria	Facultatea de Matematica si Informatica	Informatica	1
1	2	Popescu Ana	Facultatea de Matematica si Informatica	CTI	1
2	3	Andrei Cristiana	Facultatea de Matematica si Informatica	CTI	2
2	4	Luca Octavian	Facultatea de Automatica si Calculatoare	CTI	2
3	5	Mihai Florin	Facultatea de Automatica si Calculatoare	IS	1
3	6	Preda Ioana	Facultatea de Matematica si Informatica	Infomatica	1
4	7	Sasu Mihai	Facultatea de Automatica si Calculatoare	CTI	1

```
CREATE TABLE [PARTICIPANT] (
    id echipa int FOREIGN KEY REFERENCES Echipa(id echipa),
    id_participant int NOT NULL,
    NumeParticipant varchar(255) NOT NULL,
    Facultate varchar(255) NOT NULL,
    Domeniu varchar(255) NOT NULL,
    AnStudiu int NOT NULL,
    CONSTRAINT [PK_PARTICIPANT] PRIMARY KEY CLUSTERED
    (
        [id_participant] ASC
    ) WITH (IGNORE_DUP_KEY = OFF)
```

```
INSERT INTO Participant VALUES(1,1,'Anghel Maria','Facultatea de Matematica si Informatica','Informatica',1)
INSERT INTO Participant VALUES(1,2,'Popescu Ana','Facultatea de Matematica si Informatica','CTI',1)
INSERT INTO Participant VALUES(2,3,'Andrei Cristiana','Facultatea de Matematica si Informatica','CTI',2)
INSERT INTO Participant VALUES(2,4,'Luca Octavian','Facultatea de Automatica si Calculatoare','CTI',2)
INSERT INTO Participant VALUES(3,5,'Mihai Florin','Facultatea de Automatica si Calculatoare','IS',1)
INSERT INTO Participant VALUES(3,6,'Preda Ioana','Facultatea de Matematica si Informatica','Infomatica',1)
INSERT INTO Participant VALUES(4,7,'Sasu Mihai','Facultatea de Automatica si Calculatoare', 'CTI',1)
INSERT INTO Participant VALUES(4,8,'Ionescu Alin','Facultatea de Matematica si Informatica','Informatica',2)
```

```
select *
from PARTICIPANT;
```

PREMIU

```
CREATE TABLE [PREMIU] (  
id_premiu int NOT NULL,  
id_firma int FOREIGN KEY REFERENCES Firma(id_firma),  
Tip_premiu varchar(255) NOT NULL,  
SumaPremiu varchar(255) NOT NULL,
```

```
CONSTRAINT [PK_PREMIU] PRIMARY KEY CLUSTERED  
(  
[id_premiu] ASC  
) WITH (IGNORE_DUP_KEY = OFF)
```

```
)  
INSERT INTO Premiu VALUES(1,1,'Cea mai buna idee','2000 lei')  
INSERT INTO Premiu VALUES(2,1,'Cea mai rapida aplicatie','3000 lei')  
INSERT INTO Premiu VALUES(3,2,'Cea mai securizata aplicatie','3500 lei')  
INSERT INTO Premiu VALUES(4,3,'Cel mai clean code','2000 lei')  
INSERT INTO Premiu VALUES(5,4,'Cea mai buna aplicatie','3000 lei')  
INSERT INTO Premiu VALUES(6,5,'Cea mai complexa implementare','3000 lei')
```

```
select *  
from PREMIU;
```

```
CREATE TABLE [PREMIU] (  
id_premiu int NOT NULL,  
id_firma int FOREIGN KEY REFERENCES Firma(id_firma),  
Tip_premiu varchar(255) NOT NULL,  
SumaPremiu varchar(255) NOT NULL,  
  
CONSTRAINT [PK_PREMIU] PRIMARY KEY CLUSTERED  
(  
[id_premiu] ASC  
) WITH (IGNORE_DUP_KEY = OFF)  
  
)  
INSERT INTO Premiu VALUES(1,1,'Cea mai buna idee','2000 lei')  
INSERT INTO Premiu VALUES(2,1,'Cea mai rapida aplicatie','3000 lei')  
INSERT INTO Premiu VALUES(3,2,'Cea mai securizata aplicatie','3500 lei')  
INSERT INTO Premiu VALUES(4,3,'Cel mai clean code','2000 lei')  
INSERT INTO Premiu VALUES(5,4,'Cea mai buna aplicatie','3000 lei')  
INSERT INTO Premiu VALUES(6,5,'Cea mai complexa implementare','3000 lei')
```

```
select *  
from PREMIU;
```

Results Messages

id_premiu	id_firma	Tip_premiu	SumaPremiu
1	1	Cea mai buna idee	2000 lei
2	1	Cea mai rapida aplicatie	3000 lei
3	2	Cea mai securizata aplicatie	3500 lei
4	3	Cel mai clean code	2000 lei
5	4	Cea mai buna aplicatie	3000 lei
6	5	Cea mai complexa implementare	3000 lei

TRAININGFIRMA

```
CREATE TABLE [TRAININGFIRMA](
  id_firma int FOREIGN KEY REFERENCES Firma(id_firma),
  NumeTraining varchar(255) NOT NULL,
  DataTraining datetime NOT NULL,
  CONSTRAINT [PK_TRAININGFIRMA] PRIMARY KEY CLUSTERED
  (
    [id_firma] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)
)

INSERT INTO TRAININGFIRMA VALUES(2,'CybersecurityTools',convert(DATETIME, '26-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(1,'New Software Design',convert(DATETIME, '27-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(3,'Design -Creative Tools',convert(DATETIME, '28-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(4,'Secured-App',convert(DATETIME, '29-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(5,'AI & Modern World',convert(DATETIME, '30-09-2022', 103))
```

```
select *
from TRAININGFIRMA;
```

%

Results Messages

id_firma	NumeTraining	DataTraining
1	New Software Design	2022-09-27 00:00:00.000
2	CybersecurityTools	2022-09-26 00:00:00.000
3	Design -Creative Tools	2022-09-28 00:00:00.000
4	Secured-App	2022-09-29 00:00:00.000
5	AI & Modern World	2022-09-30 00:00:00.000

```
INSERT INTO TRAININGFIRMA VALUES(2,'CybersecurityTools',convert(DATETIME, '26-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(1,'New Software Design',convert(DATETIME, '27-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(3,'Design -Creative Tools',convert(DATETIME, '28-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(4,'Secured-App',convert(DATETIME, '29-09-2022', 103))
INSERT INTO TRAININGFIRMA VALUES(5,'AI & Modern World',convert(DATETIME, '30-09-2022', 103))
```

```
select *
from TRAININGFIRMA;
```

```
CREATE TABLE [TRAININGFIRMA](
  id_firma int FOREIGN KEY REFERENCES Firma(id_firma),
  NumeTraining varchar(255) NOT NULL,
  DataTraining datetime NOT NULL,
  CONSTRAINT
[PK_TRAININGFIRMA] PRIMARY KEY CLUSTERED
  (
    [id_firma] ASC
  ) WITH (IGNORE_DUP_KEY = OFF)
```

PREMIUECHIPAMAPPING

```
CREATE TABLE [PREMIUECHIPAMAPPING] (  
id_premiu int CONSTRAINT pk_p_premiu REFERENCES Premiu(id_premiu),  
id_echipa int CONSTRAINT pk_e_echipa REFERENCES Echipa(id_echipa),  
  
)
```

```
INSERT INTO PremiuEchipaMapping VALUES (1,2)  
INSERT INTO PremiuEchipaMapping VALUES (3,2)  
INSERT INTO PremiuEchipaMapping VALUES (1,3)  
INSERT INTO PremiuEchipaMapping VALUES (4,5)  
INSERT INTO PremiuEchipaMapping VALUES (2,3)  
INSERT INTO PremiuEchipaMapping VALUES (4,1)  
INSERT INTO PremiuEchipaMapping VALUES (5,1)  
INSERT INTO PremiuEchipaMapping VALUES (2,4)  
INSERT INTO PremiuEchipaMapping VALUES (3,4)  
INSERT INTO PremiuEchipaMapping VALUES (6,3)
```

```
SELECT*  
FROM PREMIUECHIPAMAPPING
```

```
CREATE TABLE [PREMIUECHIPAMAPPING] (  
    id_premiu int CONSTRAINT pk_p_premiu REFERENCES Premiu(id_premiu),  
    id_echipa int CONSTRAINT pk_e_echipa REFERENCES Echipa(id_echipa),  
)  
INSERT INTO PremiuEchipaMapping VALUES (1,2)  
INSERT INTO PremiuEchipaMapping VALUES (3,2)  
INSERT INTO PremiuEchipaMapping VALUES (1,3)  
INSERT INTO PremiuEchipaMapping VALUES (4,5)  
INSERT INTO PremiuEchipaMapping VALUES (2,3)  
INSERT INTO PremiuEchipaMapping VALUES (4,1)  
INSERT INTO PremiuEchipaMapping VALUES (5,1)  
INSERT INTO PremiuEchipaMapping VALUES (2,4)  
INSERT INTO PremiuEchipaMapping VALUES (3,4)  
INSERT INTO PremiuEchipaMapping VALUES (6,3)  
  
SELECT*  
FROM PREMIUECHIPAMAPPING
```

1 %

Results Messages

	id_premiu	id_echipa
1	1	2
2	3	2
3	1	3
4	4	5
5	2	3
6	4	1
7	5	1
8	2	4
9	3	4
10	6	3

FIRMAECHIPAMAPPING

```
CREATE TABLE [FIRMAECHIPAMAPPING] (  
    id_firma int CONSTRAINT pk_f_firma REFERENCES Firma(id_firma),  
    id_echipa int CONSTRAINT pk_c_echipa REFERENCES Echipa(id_echipa),  
)  
  
INSERT INTO FirmaEchipaMapping VALUES (1,1)  
INSERT INTO FirmaEchipaMapping VALUES (1,2)  
INSERT INTO FirmaEchipaMapping VALUES (5,1)  
INSERT INTO FirmaEchipaMapping VALUES (4,4)  
INSERT INTO FirmaEchipaMapping VALUES (4,3)  
INSERT INTO FirmaEchipaMapping VALUES (5,2)  
INSERT INTO FirmaEchipaMapping VALUES (2,5)  
INSERT INTO FirmaEchipaMapping VALUES (3,3)  
INSERT INTO FirmaEchipaMapping VALUES (3,4)  
INSERT INTO FirmaEchipaMapping VALUES (2,4)  
  
select *  
from FIRMAECHIPAMAPPING;
```

%

Results Messages

id_firma	id_echipa
1	1
1	2
5	1
4	4
4	3
5	2
2	5
3	3
3	4
2	4

```
CREATE TABLE [FIRMAECHIPAMAPPING] (  
    id_firma int CONSTRAINT pk_f_firma REFERENCES  
Firma(id_firma),  
    id_echipa int CONSTRAINT pk_c_echipa REFERENCES  
Echipa(id_echipa),  
)
```

```
INSERT INTO FirmaEchipaMapping VALUES (1,1)  
INSERT INTO FirmaEchipaMapping VALUES (1,2)  
INSERT INTO FirmaEchipaMapping VALUES (5,1)  
INSERT INTO FirmaEchipaMapping VALUES (4,4)  
INSERT INTO FirmaEchipaMapping VALUES (4,3)  
INSERT INTO FirmaEchipaMapping VALUES (5,2)  
INSERT INTO FirmaEchipaMapping VALUES (2,5)  
INSERT INTO FirmaEchipaMapping VALUES (3,3)  
INSERT INTO FirmaEchipaMapping VALUES (3,4)  
INSERT INTO FirmaEchipaMapping VALUES (2,4)
```

```
select *  
from FIRMAECHIPAMAPPING;
```

PACHETFIRMA

```
CREATE TABLE [PACHETFIRMA](
  id_firma int FOREIGN KEY REFERENCES Firma(id_firma),
  id_pachet int NOT NULL,
  facilitati varchar(255) NOT NULL,
  pret int NOT NULL,
```

```
  CONSTRAINT [PK_PACHETFIRMA]
  PRIMARY KEY CLUSTERED
```

```
(
  [id_pachet] ASC
) WITH (IGNORE_DUP_KEY = OFF)
```

```
INSERT INTO PACHETFIRMA VALUES(1,1,'număr mai mic de mentori, promovare doar pe rețele de
socializare',3000)
INSERT INTO PACHETFIRMA VALUES(2,2,'număr maxim de mentori, promovare pe rețele de socializare si
fizic',5000)
INSERT INTO PACHETFIRMA VALUES(3,3,'număr mai mic de mentori, promovare doar pe rețele de
socializare',3000)
INSERT INTO PACHETFIRMA VALUES(4,4,'număr mai mic de mentori, promovare doar pe rețele de
socializare',3000)
INSERT INTO PACHETFIRMA VALUES(5,5,'număr maxim de mentori, promovare pe rețele de socializare si
fizic',5000)
```

```
select *
from PACHETFIRMA;
```

The screenshot shows the SQL Server Enterprise Manager interface. At the top, the SQL script for creating the PACHETFIRMA table is displayed. Below the script, the 'Results' tab shows the data inserted into the table. The table has four columns: id_firma, id_pachet, facilitati, and pret. The data is as follows:

id_firma	id_pachet	facilitati	pret
1	1	numar mai mic de mentori, promovare doar pe retele...	3000
2	2	numar maxim de mentori, promovare pe retele de s...	5000
3	3	numar mai mic de mentori, promovare doar pe retele...	3000
4	4	numar mai mic de mentori, promovare doar pe retele...	3000
5	5	numar maxim de mentori, promovare pe retele de s...	5000

11. Cereri SQL



- **Cererea 1:**

Să se afișeze tipul materialului de promovare și cantitatea acestuia, care a crescut cu 50% pentru evenimentul din 2022. În această cerere am folosit NVL, filtrare la nivel de linii, subcereri nesincronizate, funcții care lucrează cu date.

```
select TipMaterial, COALESCE(NumarMaterial+0.5*NumarMaterial,0) AS
NumarMaterialFinal
from MATERIAL
WHERE id_promovare IN
(SELECT id_promovare from PROMOVARE where id_eveniment=(SELECT
id_eveniment from EVENIMENT where DATEPART (year,dataeveniment)=2022));
```

-

```
select TipMaterial, COALESCE(NumarMaterial+0.5*NumarMaterial,0) AS NumarMaterialFinal
from MATERIAL
WHERE id_promovare IN
(SELECT id_promovare from PROMOVARE where id_eveniment=(SELECT id_eveniment from EVENIMENT where DATEPART (year,dataeveniment)=2022));
```

results Messages

TipMaterial	NumarMaterialFinal
Afise	15.0
Flyere	150.0

- **Cererea 2:**

Să se afișeze numele echipelor și numărul participanții care câștigă premiul în valoare de 3500 de lei. Dacă acesta va fi mai mic decât 4, vom afișa că echipa este incompletă, iar dacă este 4, atunci echipa e completă. Pentru această cerere am folosit funcții pe șiruri, case, filtrare la nivel de linii, subcereri nesincronizate.

```
Select e.Nume_echipa, e.NumarParticipanti,  
case when e.NumarParticipanti<4 then 'Echipa cu mai putini de 4 membri'  
when e.NumarParticipanti=4 then 'Echipa cu 4 membri'  
end  
as MembriEchipa  
FROM ECHIPA e  
WHERE id_echipa in  
(SELECT id_echipa from PREMIUECHIPAMAPPING where id_premiu=(SELECT id_premiu from PREMIU  
where CAST(LEFT(SumaPremiu, 4) AS int)=3500))
```

```
Select e.Nume_echipa, e.NumarParticipanti,  
case when e.NumarParticipanti<4 then 'Echipa cu mai putini de 4 membri'  
when e.NumarParticipanti=4 then 'Echipa cu 4 membri'  
end  
as MembriEchipa  
FROM ECHIPA e  
WHERE id_echipa in  
(SELECT id_echipa from PREMIUECHIPAMAPPING where id_premiu=(SELECT id_premiu from PREMIU where CAST(LEFT(SumaPremiu, 4) AS int)=3500));
```

Results Messages

Nume_echipa	NumarParticipanti	MembriEchipa
Nu!Team	4	Echipa cu 4 membri
FANTASTIC 4	4	Echipa cu 4 membri

- **Cererea 3:**

Să se afișeze numele trainingurilor, ale firmelor care le țin și ale echipelor participante, grupate întâi după numele firmei, apoi al echipei și în final al trainingului, iar funcția count ne asigură că fiecare echipă participă la traininguri, dacă se află la o dată diferită. Am folosit join pe 4 tabele, grupări de date, funcții grup și clauza with.

```
with TABLETEMP as(SELECT FIRMA.NumeFirma FROM FIRMA)
SELECT count(*) as
TotalTraining,TRAININGFIRMA.NumeTraining,
ECHIPA.Nume_echipa
FROM TRAININGFIRMA
INNER JOIN FIRMA ON TRAININGFIRMA.id_firma =
FIRMA.id_firma
INNER JOIN FIRMAECHIPAMAPPING ON FIRMA.id_firma =
FIRMAECHIPAMAPPING.id_firma
INNER JOIN ECHIPA ON FIRMAECHIPAMAPPING.id_echipa=
ECHIPA.id_echipa
GROUP BY FIRMA.NumeFirma,ECHIPA.Nume_echipa,
TRAININGFIRMA.NumeTraining
```

```
with TABLETEMP as(SELECT FIRMA.NumeFirma FROM FIRMA)
SELECT count(*) as TotalTraining,TRAININGFIRMA.NumeTraining,
        ECHIPA.Nume_echipa
FROM TRAININGFIRMA
INNER JOIN FIRMA ON TRAININGFIRMA.id_firma = FIRMA.id_firma
INNER JOIN FIRMAECHIPAMAPPING ON FIRMA.id_firma = FIRMAECHIPAMAPPING.id_firma
INNER JOIN ECHIPA ON FIRMAECHIPAMAPPING.id_echipa= ECHIPA.id_echipa
GROUP BY FIRMA.NumeFirma,ECHIPA.Nume_echipa, TRAININGFIRMA.NumeTraining
```

TotalTraining	NumeTraining	Nume_echipa
1	AI & Modern World	console.log(Win)
1	AI & Modern World	NullTeam
1	Design -Creative Tools	FANTASTIC 4
1	Design -Creative Tools	hackU
1	CybersecurityTools	Computer Mechanics
1	CybersecurityTools	FANTASTIC 4
1	Secured-App	FANTASTIC 4
1	Secured-App	hackU
1	New Software Design	console.log(Win)
1	New Software Design	NullTeam

- **4. Cererea 4**

Presupunem ca vrem sa aflam: Care este numele, domeniul si anul de studii persoanelor care au facut parte din echipa care a castigat al doilea premiu, deoarece au plecat inainte de premiere. Pentru a afla asta, vom folosi in query, o cerere sincronizata pe 3 tabele, functia concat.

```
SELECT CONCAT(CONCAT(p.Facultate, ' '),p.Domeniu)+STR(p.AnStudiu) as "Facultate si Domeniu",p.id echipa as "Echipa", p.NumeParticipant as "Nume"
from PARTICIPANT p
where p.id_participant in (select id_participant
                           from PARTICIPANT
                           where id echipa > (select avg(id echipa)
                                                from Echipa))
and p.id_participant in(select id_participant
                        from PARTICIPANT
                        where id echipa in (select id echipa
                                           from PREMIUECHIPAMAPPING
                                           where id_premiu > 2
                                           )
                        );
```

Results Messages

Facultate si Domeniu	Echipa	Nume
Facultatea de Automatica si Calculatoare CTI ...	4	Sasu Mihai
Facultatea de Matematica si Informatica Informat...	4	Ionescu Alin

- 5. Cererea 5

Voluntarii aflati intr-o tura afla ca a sosit un sponsor, iar unul dintre ei vede produsele si le transmite celorlalti printr-o codificare.

Astfel, fiecare produs va primi o noua denumire si vor fi date participantilor in ordine descrescatoare a numarului lor.

Pentru acest lucru vom folosi ordonarea produselor, decodificarea si o functie de grup.

```
SELECT NumarProduce, NumeSponsor,
```

```
CASE TipProduce
```

```
    WHEN 'Pizza' THEN 'P'
```

```
    WHEN 'Suc fresh' THEN 'S'
```

```
    WHEN 'Energizant' THEN 'E'
```

```
    WHEN 'Apa' THEN 'A'
```

```
    WHEN 'Salata' THEN 'F-F'
```

```
    WHEN 'Snacksuri' THEN 'SK'
```

```
    ELSE 'Alt produs'
```

```
END
```

```
FROM SPONSOR
```

```
ORDER BY NumarProduce DESC;
```

```
SELECT SUM (NumarProduce) AS "Numar total produse"
```

```
FROM SPONSOR
```

```
SELECT NumarProduce, NumeSponsor,  
CASE TipProduce  
    WHEN 'Pizza' THEN 'P'  
    WHEN 'Suc fresh' THEN 'S'  
    WHEN 'Energizant' THEN 'E'  
    WHEN 'Apa' THEN 'A'  
    WHEN 'Salata' THEN 'F-F'  
    WHEN 'Snacksuri' THEN 'SK'  
    ELSE 'Alt produs'  
END  
FROM SPONSOR  
ORDER BY NumarProduce DESC;  
  
SELECT SUM (NumarProduce) AS "Numar total produse"  
FROM SPONSOR
```

Results Messages

NumarProduce	NumeSponsor	(No column name)
400	Aqua Carpatica	A
200	Sloop	S
200	Red Bull	E
150	Chio	SK
100	TrentaPizza	P
100	Fru Fru	F-F

Numar total produse

1150

12. Operații de actualizare/ suprimare a datelor utilizând subcereri.



1. Două firme partenere (cele cu id par) doresc să mărească suma investită într-un pachet cu 20%. Să se construiască o cerere care permite acest lucru.

```
update PACHETFIRMA
set pret = pret+0.2*pret
where id_pachet in (select id_pachet from
                    PACHETFIRMA
                    where id_firma in (select id_firma from FIRMA where
                    (id_firma%2) <1));
```

```
update PACHETFIRMA
set pret = pret+0.2*pret
where id_pachet in (select id_pachet from PACHETFIRMA
                    where id_firma in (select id_firma from FIRMA where (id_firma%2) <1));
```

2 rows affected)

```
SELECT pret
FROM PACHETFIRMA
```

pret
3000
6000
3000
3600
5000

Inițial pentru 2 și 4 prețurile erau 5000, respectiv 3000, deci s-au modificat corespunzător în urma cererii

12. Operații de actualizare/ suprimare a datelor utilizând subcereri.



2. Pentru că în anii 2021 și 2020 evenimentul s-a desfășurat în format online, se dorește alcătuirea unei cereri care să reducă la jumătate numărul materialelor de promovare.

update MATERIAL

set NumarMaterial=NumarMaterial-0.5*NumarMaterial

where id_promovare in (select id_promovare from PROMOVARE

where id_eveniment in(select id_eveniment from EVENIMENT where DATEPART (year,dataeveniment)=2021 OR DATEPART(year,dataeveniment)=2020));

```
update MATERIAL
set NumarMaterial=NumarMaterial-0.5*NumarMaterial
where id_promovare in (select id_promovare from PROMOVARE
where id_eveniment in(select id_eveniment from EVENIMENT where DATEPART (year,dataeveniment)=2021 OR DATEPART(year,dataeveniment)=2020 ));
```

%

Messages

(2 rows affected)

12. Operații de actualizare/ suprimare a datelor utilizând subcereri.



3. Să se elimine din materialele de promovare Cănille.

`select*from MATERIAL;`

`delete from MATERIAL`

`WHERE TipMaterial = 'Cani'`

`select*from MATERIAL;`

```
select*from MATERIAL;

delete from MATERIAL
WHERE TipMaterial = 'Cani'

select*from MATERIAL;
```

%

Results Messages

id_promovare	TipMaterial	NumarMaterial
1	Afise	10
2	Flyere	100
3	Pixuri	25
4	Tricouri	37
5	Cani	50

id_promovare	TipMaterial	NumarMaterial
1	Afise	10
2	Flyere	100
3	Pixuri	25
4	Tricouri	37

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



```
CREATE SEQUENCE sequence_1  
start with 2  
increment by 2  
minvalue 0  
maxvalue 100  
cycle;
```

```
CREATE SEQUENCE sequence_2  
start with 1  
increment by 3  
minvalue 0  
maxvalue 100  
cycle;
```

14. Crearea unei vizualizări compuse.



Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

```
CREATE VIEW
VIEW TRAININGFIRMA FIRMA AS
(
    SELECT DISTINCT
        T.id_firma,
        T.NumeTraining,
        F.NumeFirma,
        F.NumeTemaPropusa
    FROM TRAININGFIRMA T
    JOIN FIRMA F on T.id_firma = F.id_firma
    WHERE (LEN(T.NumeTraining) > 10)
)

--Operatie permisă
SELECT* FROM VIEW_TRAININGFIRMA_FIRMA ORDER BY id_firma

--Operatie nepermisa
--DELETE FROM VIEW_TRAININGFIRMA_FIRMA
--WHERE id_firma = 2
```

id_firma	NumeTraining	NumeFirma	NumeTemaPropusa
1	New Software Design	Softbinator	AI in Health
2	CybersecurityTools	Bitdefender	Application simulating an Operating S
3	Design -Creative Tools	Adobe	Mini-Creative Cloud Application
4	Secured-App	Luxoft	Capture the flag
5	AI & Modern World	Accenture	The art of AI-application

Am creat vizualizarea compusă formată din attributele ale celor 2 tabele: TRAININGFIRMA și FIRMA. Asupra acesteia putem efectua operații de afișare, dar nu putem modifica datele, pentru că în vizualizarea dată operăm cu 2 tabele de bază.

```
DELETE FROM VIEW_TRAININGFIRMA_FIRMA
WHERE id_firma = 2
```

Messages

Msg 4405, Level 16, State 1, Line 477
View or function 'VIEW_TRAININGFIRMA_FIRMA' is not updatable because the modification affects multiple base tables.

15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii



Un index nou creat pentru attributele unei entități este folosit pentru a putea verifica mai eficient 2 attribute în funcție de care vrem să facem o cerere. În exemplul dat, dorim să verificăm pentru un participant la concurs dacă este student la Facultatea de Matematică și Informatică, domeniul Informatică, folosind indexul `ix_participant`.

```
CREATE INDEX ix_participant
ON PARTICIPANT(Facultate, Domeniu);
SELECT
    NumeParticipant,
    Facultate,
    Domeniu
FROM
    PARTICIPANT
WHERE
    Facultate = 'Facultatea de Matematica si Informatica' AND
    Domeniu = 'Informatica';
```

%

Results Messages

NumeParticipant	Facultate	Domeniu
Anghel Maria	Facultatea de Matematica si Informatica	Informatica
Ionescu Alin	Facultatea de Matematica si Informatica	Informatica

16. Cerere ce utilizează operația outer-join pe minimum 4 tabele și două cereri ce utilizează operația division.



1.Outer Join pe 4 tabele

```
SELECT PREMIU.Tip_premiu, PARTICIPANT.NumeParticipant,
        ECHIPA.Nume echipa
FROM PARTICIPANT
LEFT OUTER JOIN ECHIPA ON PARTICIPANT.id_echipa = ECHIPA.id_echipa
LEFT OUTER JOIN PREMIUECHIPAMAPPING ON ECHIPA.id_echipa = PREMIUECHIPAMAPPING.id_echipa
FULL OUTER JOIN PREMIU ON PREMIUECHIPAMAPPING.id_premiu= PREMIU.id_premiu
GROUP BY PREMIU.Tip_premiu,PARTICIPANT.NumeParticipant, ECHIPA.Nume echipa
```

Tip_premiu	NumeParticipant	Nume echipa
Cea mai buna aplicatie	Anghel Maria	console.log(Win)
Cea mai buna aplicatie	Popescu Ana	console.log(Win)
Cea mai buna idee	Andrei Cristiana	NullTeam
Cea mai buna idee	Luca Octavian	NullTeam
Cea mai buna idee	Mihai Florin	hackU
Cea mai buna idee	Preda Ioana	hackU
Cea mai complexa implementare	Mihai Florin	hackU
Cea mai complexa implementare	Preda Ioana	hackU
Cea mai rapida aplicatie	Ionescu Alin	FANTASTIC 4
Cea mai rapida aplicatie	Mihai Florin	hackU
Cea mai rapida aplicatie	Preda Ioana	hackU
Cea mai rapida aplicatie	Sasu Mihai	FANTASTIC 4
Cea mai securizata aplicatie	Andrei Cristiana	NullTeam
Cea mai securizata aplicatie	Ionescu Alin	FANTASTIC 4
Cea mai securizata aplicatie	Luca Octavian	NullTeam
Cea mai securizata aplicatie	Sasu Mihai	FANTASTIC 4
Cel mai clean code	Anghel Maria	console.log(Win)
Cel mai clean code	Popescu Ana	console.log(Win)

Afișați, pentru fiecare participant din ce echipă face parte și toate premiile posibile obținute de echipă.

Cuvântul cheie **FULL OUTER JOIN** returnează toate înregistrările atunci când există o potrivire în înregistrările tabelului din stânga sau din dreapta .

Cuvântul cheie **LEFT JOIN** returnează toate înregistrările din tabelul din stânga și înregistrările care se potrivesc din tabelul din dreapta .

2. Prima cerere division



Să se afișeze id-ul echipelor și numele celor care au câștigat un premiu cel puțin la fel de bun ca echipa cu id-ul 4.

```
SELECT e.id echipa, Nume_echipa
```

```
FROM ECHIPA e JOIN PREMIUECHIPAMAPPING pe ON (e.id echipa = pe.id echipa),  
PREMIUECHIPAMAPPING JOIN PREMIU p ON (PREMIUECHIPAMAPPING.id_premiu= p.id_premiu)
```

```
WHERE p.SumaPremiu > '3000 lei' AND e.id echipa!=4  
ORDER BY e.id echipa
```

```
SELECT e.id echipa, Nume_echipa  
  
FROM ECHIPA e JOIN PREMIUECHIPAMAPPING pe ON (e.id echipa = pe.id echipa),  
PREMIUECHIPAMAPPING JOIN PREMIU p ON (PREMIUECHIPAMAPPING.id_premiu= p.id_premiu)  
  
WHERE p.SumaPremiu > '3000 lei' AND e.id echipa!=4  
ORDER BY e.id echipa
```

id_echipa	Nume_echipa
1	console.log(Win)
1	console.log(Win)
1	console.log(Win)
1	console.log(Win)
2	NullTeam
2	NullTeam
2	NullTeam
2	NullTeam
3	hack11

3. A doua cerere division



Să se afișeze id-ul, numele participanților care au obținut premiul de 2000 lei.

```
SELECT DISTINCT c.NumeParticipant, c.id_participant
```

```
FROM PARTICIPANT c JOIN ECHIPA e ON (e.id_echipa = c.id_echipa),  
ECHIPA JOIN PREMIUECHIPAMAPPING pe ON (ECHIPA.id_echipa = pe.id_echipa),  
PREMIUECHIPAMAPPING JOIN PREMIU p ON (PREMIUECHIPAMAPPING.id_premiu= p.id_premiu)
```

```
WHERE p.SumaPremiu = '2000 lei'
```

```
ORDER BY c.id_participant
```

```
--SELECT DISTINCT c.NumeParticipant, c.id_participant  
  
FROM PARTICIPANT c JOIN ECHIPA e ON (e.id_echipa = c.id_echipa),  
ECHIPA JOIN PREMIUECHIPAMAPPING pe ON (ECHIPA.id_echipa = pe.id_echipa),  
PREMIUECHIPAMAPPING JOIN PREMIU p ON (PREMIUECHIPAMAPPING.id_premiu= p.id_premiu)  
  
WHERE p.SumaPremiu = '2000 lei'  
ORDER BY c.id_participant
```

6

Results Messages

NumeParticipant	id_participant
Anghel Maria	1
Popescu Ana	2
Andrei Cristiana	3
Luca Octavian	4
Mihai Florin	5
Preda Ioana	6
Sasu Mihai	7
Ionescu Alin	8

18.Normalizarea BCNF, FN4, FN5 + Denormalizarea.



- **Forma normală Boyce-Codd (BCNF)**

Definiția acestei forme normale este următoarea. O relație se află în forma normală mai sus menționată dacă și numai dacă fiecare determinant este o cheie candidat.

Pentru a exemplifica o normalizare de tipul BCNF se va crea un exemplu fictiv în care presupunem că atributul Tip_Premiu este unic (să zicem că ar fi premiile I,II și III) și determină funcțional atributul id_premiu# care face parte din cheia primară. (acest exemplu este creat doar pentru rezolvarea acestui exercițiu, deoarece aceasta ar restricționa numărul de premii și ar impune un mod de a le denumi, diferit de cel ales în model.

În acest mod, între attributele relației Tip_premiu_id_premiu_id_firma : (id_premiu#, id_firma#, Tip_premiu) există dependențele:

o {id_premiu#, id_firma#}-> {Tip_premiu},

o {Tip_premiu}->{id_premiu#}

Se poate observa faptul că Tip_premiu este o cheie candidat (pentru că este unică).

Totodată, cheia id_premiu depinde de atributul Tip_premiu .

Se aplică regula Casey-Delobel și se aduce relația în BCNF => tabela se va **împărți în două**: una cu id_firma#, Tip_premiu și cealaltă cu id_premiu#, Tip_premiu.



- **Forma normala 4 (FN4)**

O relație se află în a patra formă normală dacă și numai dacă se află în forma normală Boyce-Codd și nu conține relații m:n independente (orice dependență multivaloare este o dependență funcțională)

Fie următorul exemplu: în entitatea LOCATIE (id_locatie#, NumeLocatie, ContactLocatie, NumarCamereDisponibile) o locație poate avea mai multe denumiri (să zicem că primește denumiri diferite în funcție de intrările prin care poți ajunge la ea) și mai multe date de contact. Astfel, dacă se va dori să se introducă o nouă locație va trebui să se introducă și numele și contactul.

Relația LOCATIE este în BCNF. Pentru a aduce relația în FN4 o vom descompune prin proiecție în două relații: **LOCATIE1 (id_locatie#, NumeLocatie)**, **LOCATIE2(id_locatie#, ContactLocatie) ,**
LOCATIE = JOIN(LOCATIE1, LOCATIE2).



- **Forma normală 5 - FN5**

FN5 își propune eliminarea redundanțelor care apar în relații m:n dependente. Intuitiv, o relație R este în forma normală 5 dacă și numai dacă este în FN4, iar aceasta nu conține dependențe ciclice. Acest lucru este intuitiv, deoarece fiecare dependență multivaloare poate fi privită ca un caz particular de dependență de uniune.

Orice relație poate fi descompusă fără pierderi la uniune într-o mulțime de relații care sunt în FN5. Pentru a preciza dacă o relație este în FN5, este suficient să cunoaștem cheile candidate și toate dependențele de uniune din relație. Aducerea în FN5 presupune eliminarea join dependențelor.

Fie următorul exemplu din modelul dat: O firmă acordă un premiu/ mai multe unei echipe în funcție de performanța acesteia, iar un tip de premiu se poate acorda la mai multe echipe.

Se poate remarca faptul că cele trei relații de tip 2 au compus o diagramă care conține dependențe ciclice, ceea ce înseamnă că relația precizată mai sus nu este în FN5. Mai mult, relația fiind ciclică, aceasta va avea o relație de tip 3 echivalentă. Astfel, în urma normalizării (eliminarea ciclului) diagrama va avea încă o relație PREMIUECHIPAMAPPING, ajungând să fie o relație de tip 3.

• Denormalizare



Obiectivul denormalizării constă în reducerea numărului de join-uri care trebuie efectuate pentru rezolvarea unei interogări, prin realizarea unora dintre acestea în avans, ca făcând parte din proiectarea bazei de date. Ca o regulă se poate afirma că, dacă performanțele nu sunt satisfăcătoare și relația are o rată de reactualizare scăzută, dar o rată a interogărilor foarte ridicată, denormalizarea poate constitui o opțiune viabilă.

Se consideră următorul exemplu: avem un tabel în care sunt stocate metodele de promovare ale unui eveniment, numit PROMOVARE. Metodelor clasice de promovare online li se adaugă și materialele de promovare (relație 1:M dintre entitatea PROMOVARE și MATERIAL).

Astfel, în urma denormalizării, atributele din entitatea MATERIAL vor fi plasate în tabelul PROMOVARE, pentru că în acest fel se elimină o operație de join care este costisitoare. În acest mod diagrama este mai eficientă, iar timpul interogărilor a fost îmbunătățit.

Denormalizarea a fost necesară pentru că nu era eficient ca atributele TipMaterial și NumarMaterial să se afle într-un tabel separat față de tabelul PROMOVARE, având în vedere că această poziționarea implica un join suplimentar pentru a accesa cele două date din tabelul MATERIAL. Așa cum s-a precizat mai sus, join-urile sunt costisitoare și de aceea trebuia redus numărul lor.