

Platforma de invatare online

Ilies Andrei-Cristian, Macovei Alexandru-Fabian

1 Descrierea cerintelor sistemului

1. O platforma de invatare online contine mai multe cursuri deschise publicului, contra cost sau gratuit. Acestea sunt impartite pe domenii si sunt realizate de creatori, certificati pentru a preda pe domeniul cursului creat/cursurilor create. Cursurile se desfasoara individual si pot fi parcurse in propriul ritm.
2. Un mentor este un angajat al platformei de invatare. Acesta poate fi specializat pe mai multe domenii si trebuie sa ajute participantii. Fiecare curs poate avea unul sau mai multi mentori, si fiecare mentor poate activa in unul sau mai multe cursuri.
3. Un creator este o persoana care a alcatuit unul sau mai multe cursuri aprobate de catre mentori. Acesta primeste un procent din toate vanzarile cursurilor. Creatorul are dreptul de a raspunde la intrebari cu privire la cursurile create, dar nu este obligat sa o faca.
4. Un cursant poate sa se inscrie la oricate cursuri. Acestia pot adresa intrebari catre mentorii cursului, intrebari care vor putea fi vazute de ceilalti participanti ai cursului oricand.
5. La sfarsitul unui curs, utilizatorul va trebui sa alcatuiasca un proiect. Proiectul va fi verificat de unul dintre mentori, care va stabili daca participantul este sau nu eligibil la certificarea pentru cursul respectiv. Proiectele respinse pot fi refacute gratuit o singura data, dupa care trebuie reluat cursul contra cost.

2 Proiectarea logica a bazei de date

La aceasta etapa va fi descris procesul de construire a schemei logice pentru baza de date "Platforma de invatare online".

2.1 Identificarea tipurilor de entitati

1. Creator
2. Mentor

3. Cursant
4. Administrator
5. Curs
6. Proiect final
7. Domeniu
8. Recenzie curs
9. Conversatie
10. Mesaj
11. Rol
12. Permisiiune

2.2 Identificarea tipurilor de relatii

2.2.1 Relatii de grad 2

Tip entitate	Relatie	Tip entitate	Cardinalitate
Creator	creeza	Cursuri	1:NC
	este specializat pe	Domenii	MC:NC
	are	Rol	NC:1
Mentor	este specializat pe	Domenii	MC:NC
	are	Rol	NC:1
Cursant	detine	Cursuri	MC:NC
	realizeaza	Proiecte finale	1:NC
	are	Rol	NC:1
Administrator	are	Rol	NC:1
Curs	are	Domeniu	NC:1
Mesaj	face parte din	Conversatie	N:1
	are	autor ¹	NC:1
Recenzia	este scrisa de	autor	NC:1
Rol	are	Permisiiuni	M:N

¹poate fi Mentor, Creator, Cursant sau Administrator

2.2.2 Relatii de grad n

In aceasta baza de date exista o asociere ternara intre **Curs**, **Cursant** si **Proiect final**, deoarece fiecare cursant se poate inscrie de mai multe ori la acelaasi curs (daca nu reuseste sa treaca din prima). Aceasta relatie se va numi **Instanta_curs** si va lega 1 Cursant de 1 Curs si de 2 Proiecte finale. Ca attribute, se gasesc data_inscrierii (Date), data_finalizarii (Date), status ('in curs', 'absolvit' sau 'neabsolvit' - Varchar(10)).

Diagrama relatie:

2.3 Determinarea atributelor si a domeniilor de definiție

Tipurile atributelor si domeniile lor de definitie au fost concepute pentru a optimiza memoria utilizata, dar fara a limita datele. Cateva explicatii pentru alegerile facute:

1. Atributele reprezentand denumiri au fost limitate la un numar de caractere pe care l-am considerat a fi destul de mare pentru datele care ar putea fi memorate in acestea.
2. Pentru datele de tipul sirului de caractere pentru care nu am putut vedea o limita care sa permita flexibilitatea pe care o cautam s-a folosit tipul de date "Text", care permite, din punct de vedere al utilizatorului bazei de date, introducerea a oricate caractere in atribut. Acest tip de date este echivalent cu tipul "Varchar", alegerea denumirii fiind facuta doar pentru a imbunatati putin lizibilitatea datelor din tabelul de mai jos.

Tip entitate	Atribut	Domeniu de definitie
Creator	id_creator	Integer
	nume	Varchar(30)
	prenume	Varchar(30)
	data_nasterii	Date
	adresa_email	Varchar(50)
	parola_cont	Varchar(50)
	data_alaturarii	Date
	id_rol	Integer
	procent	Numeric
	numar_telefon	Varchar(15)
Mentor	id_mentor	Integer
	nume	Varchar(30)
	prenume	Varchar(30)
	data_nasterii	Date
	adresa_email	Varchar(50)
	parola_cont	Varchar(50)
	data_angajarii	Date
	salariu	Numeric

	id_rol	Integer
	numar_telefon	Varchar(15)
Cursant	id_cursant	Integer
	nume	Varchar(30)
	prenume	Varchar(30)
	data_nasterii	Date
	adresa_email	Varchar(50)
	parola_cont	Varchar(50)
	id_rol	Integer
	numar_telefon	Varchar(15)
	data_alaturarii	Date
Administrator	id_administrator	Integer
	nume	Varchar(30)
	prenume	Varchar(30)
	data_nasterii	Date
	adresa_email	Varchar(50)
	parola_cont	Varchar(50)
	data_alaturarii	Date
	id_rol	Integer
	numar_telefon	Varchar(15)
Domeniu	id_domeniu	Integer
	denumire	Varchar(100)
Curs	id_curs	Integer
	id_creator	Integer
	id_domeniu	Integer
	denumire	Varchar(30)
	data_crearii	Date
	pret	Numeric
	link	Text
Conversatie	id_conversatie	Integer
	subiect	Varchar(100)
Mesaj	id_mesaj	Integer
	id_conversatie	Integer
	id_autor	Integer
	data_postarii	Date
	continut	Text
Proiect final	id_proiect	Integer
	continut	Text
	status	Varchar()
Recenzie curs	id_recenzie	Integer
	nota	Integer
	specificatii	Text
Rol	id_rol	Integer
	denumire	Varchar(20)
Permisiune	id_permisiune	Integer

	denumire	Varchar(30)
--	----------	-------------

2.4 Generalizarea tipurilor de entitati

Dupa cum se poate observa foarte usor, exista date comune pentru cele 4 tabele care modeleaza persoane: **Cursant**, **Mentor**, **Creator**, **Administrator**. De aceea vom extrage aceste date intr-un tip de entitate separat, care se va mosteni in fiecare dintre aceste tipuri de entitati deja existente.

Astfel, noul tip de entitate **Utilizator** va contine urmatoarele date:

- id_utilizator (un identificator unic)
- nume
- prenume
- data_nasterii
- data_alaturarii (corespunzatoarea datei angajarii din **Mentor**)
- adresa_email
- parola_cont
- numar_telefon
- id_rol

De asemenea, fiecare tip de entitate care modeleaza o persoana va avea o relatie de tipul "este un" (cardinalitate 1:C) cu **Utilizator**. Relatiile dintre **Mesaj** si autor, respectiv **Recenzie_curs** si autor se vor transforma in relatii cu **Utilizator**.

Observam, de asemenea, ca tipul de entitate Administrator a devenit o tabela goala, care nu are informatii utile si nici relatii cu alte tabele in afara de Utilizator. Din acest motiv putem elimina tipul de entitate Administrator.

Schimbarile vor arata astfel:

Tip entitate	Relatie	Tip entitate	Cardinalitate
Cursant	este un	Utilizator	1:C
Mentor	este un	Utilizator	1:C
Creator	este un	Utilizator	1:C
Utilizator	are	Rol	NC:1
Mesaj	este scris de	Utilizator	NC:1
Recenzie_curs	este scrisa de	Utilizator	NC:1

Tabelul de atribute ale tipurilor de entitati care contine doar modificarile anterioare:

Tip entitate	Atribut	Domeniu de definitie
Utilizator	id_utilizator	Integer
	nume	Varchar(30)
	prenume	Varchar(30)
	data_nasterii	Date
	data_alaturarii	Date
	adresa_email	Varchar(50)
	parola_cont	Varchar(50)
	numar_telefon	Varchar(15)
Creator	id_creator	Integer
	id_utilizator	Integer
	procent	Numeric
Mentor	id_mentor	Integer
	id_utilizator	Integer
	salariu	Numeric

2.5 Determinarea atributelor care compun cheile candidate și primare

Cu toate ca unele tipuri entitati au deja un atribut sau set de atribute care ar putea sa reprezinte unic o entitate in tabel, am preferat sa adaugam pentru toate tipurile de entitati un identificator unic separat. Acesta ne va ajuta sa intelegem mai bine relatiile dintre tipurile de entitati si sa proiectam o baza de date consistenta si mai usor de folosit. Intelegem ca prin aceasta decizie este introdus un oarecare nivel de redundanta in baza de date, dar consideram ca avantajele aduse sunt mai mari decat dezavantajele produse.

Tip entitate	Chei candidat	Cheie primara	Chei alternante
Utilizator	id_utilizator email	id_utilizator	email
Creator	id_creator id_utilizator	id_creator	id_utilizator
Mentor	id_mentor id_utilizator	id_mentor	id_utilizator
Curs	id_curs denumire	id_curs	denumire
Domeniu	id_domeniu denumire	id_domeniu	denumire
Proiect final	id_proiect	id_proiect	-

Conversatie	id_conversatie	id_conversatie	-
Mesaj	id_mesaj	id_mesaj	-
Recenzie_curs	id_recenzie	id_recenzie	-
Rol	id_rol denumire	id_rol	denumire
Permisune	id_permisiune denumire	id_permisiune	denumire

2.6 Desenarea diagramei Entity-Relationship

2.7 Specificatii cu vedere la procesul de creare a schemei logice

Initial au fost considerate mai multe tipuri de entitati decat enumerate anterior. Entitatile de tipul **Sarcina**, **Material** si **Exercitiu** au fost eliminate cu presupunerea ca un **Curs** exista online la un anumit link si este complet. Tipul de entitate **Inscriere**, care ar fi legat **Cursant** de **Curs**, a fost transformat intr-o relatie ternara intre **Cursant**, **Curs**, si **Proiect_final**.

3 Normalizarea bazei de date

O tabela este in forma normala 1 (FN1) daca fiecare atribut al acestuia este atomic, adica nu exista valori multiple sau compuse.

O tabela se afla in formala normala 2 (FN2) daca se afla in FN1 si fiecare atribut ofera informatii doar despre cheia primara (sau una dintre componentele acesteia). Aceasta se mai poate descrie ca lipsa existentei dependentelor partiale.

O tabela se gaseste in forma normala 3 (FN3) daca se afla in FN2 si nu contine dependente tranzitive, adica attributele nu depind de attribute care nu sunt cheie primara.

3.1 Exemplul 1

Pentru un prim exemplu de normalizare la FN1 vom considera relatia reprezentata prin tabela T1:

id_curs	id_creat	nume	id_dom	dom	pret	link	id_rec	nota	specif
5	12	Alex	2, 7	IA, BD	14	[link 5]	4	7	ok
							5	10	perfect
8	9	Andrei	2	IA	12	[link 8]	15	5	slab
							16	6	-
							18	8	bun
9	3	David	3, 7	AG, BD	6	[link 9]	17	3	rau

Cu toate ca baza de date proiectata anterior accepta 1 singur Domeniu pe Curs (relatie 1:N), consideram ca relatia dintre Curs si Domeniu este de M:N. Acest considerent ne obliga sa normalizam si pentru FN2. Relatia dintre Curs si Recenzie este de 1:N. Fiecare curs are 1 singur creator deci relatia Curs-Creator este de N:1.

3.1.1 Aducerea la FN1

Observam ca aceasta relatie nu se afla in FN1 deoarece exista valorile multiple posibile in attributele id_recenzie, nota si specificatii. Pentru a ne asigura ca relatia contine doar valori atomice vom sparge tabela in T1 2 tabele: tabela T2 (care contine primele 7 attribute) si tabela T3 (care contine ultimele 3 attribute si, in plus, id_curs).

Tabela T2:

id_curs	id_creator	nume	id_domeniu	denumire	pret	link
5	12	Alex	2, 7	IA, BD	14	[link 5]
8	9	Andrei	2	IA	12	[link 8]
9	3	David	3, 7	AG, BD	6	[link 9]

Tabela T3:

id_recenzie	id_curs	nota	specificatii
4	5	7	ok
5	5	10	perfect
15	8	7	ok
16	8	6	-
18	8	8	bun
17	9	3	rau

In tabela T3, id_curs este cheie straina.

In continuare vom rupe tabela T2 in 2 tabele deoarece aceasta nu se afla inca in FN1.

Tabela T4:

id_curs	id_creator	nume	pret	link
5	12	Alex	14	[link 5]
8	9	Andrei	12	[link 8]
9	3	David	6	[link 9]

Tabela T5:

id_curs	id_domeniu	denumire
5	2	IA
5	7	BD
8	2	IA
9	3	AG
9	7	BD

Vedem acum ca toate attributele contin doar valori atomice. Acest fapt ne spune ca relatia se afla in FN1.

3.1.2 Aducerea la FN2

In continuare vom aduce relatia T1 la FN2. Tabelele cu care o sa lucram sunt T3, T4 si T5. Pentru aceasta trebuie sa identificam dependentele partiale de cheia primara.

In tabelele T3 si T4 nu exista dependente partiale, deoarece cheile primare sunt singulare (id_recenzie, respectiv id_curs). Acestea vor ramane, asadar, nemodificate.

In tabela T5 observam o dependenta partiala fata de cheia primara compusa (id_curs, id_domeniu), si anume ca atributul denumire depinde doar de id_domeniu. Pentru a normaliza in continuare tabela T5 o vom sparge in 2: T6 si T7.

Tabela T6:

id_domeniu	denumire
2	IA
3	AG
7	BD

Tabela T7:

id_curs	id_domeniu
5	2
5	7
8	2
9	3
9	7

Nu mai exista dependente partiale fata de chei primare, ceea ce inseamna ca toate tabelele se afla in acest moment in FN2.

3.1.3 Aducerea la FN3

Incercam normalizarea la FN3 a tabelor T3, T4, T6 si T7.

Observam cu usurinta ca tabelele T3, T6 si T7 se afla deja in FN3, deoarece toate attributele depind de cheia primara respectiva. Acestea vor ramane neschimbate in etapa urmatoare.

In tabela T4 exista, in schimb, o dependenta tranzitiva, si anume dependenta atributului nume fata de id_domeniu. Coloana id_creator depinde mai departe de cheia primara id_curs, deci aceasta este o dependenta tranzitiva. Pentru a o rezolva, vom sparge tabela T4 in 2 tabele: T8 si T9.

Tabela T8:

id_curs	id_creator	pret	link
5	12	14	[link 5]
8	9	12	[link 8]
9	3	6	[link 9]

Tabela T9:

id_creator	nume
12	Alex
9	Andrei
3	David

3.1.4 Rezultatul final

Relatia initiala:

id_curs	id_creat	id_dom	dom	data	pret	link	id_rec	nota	specif
5	12	7	BD	22-7-5	14	[link 5]	4	7	ok
							5	10	perfect
8	9	2	IA	23-2-7	12	[link 8]	4	7	ok
							16	6	-
							18	8	bun

Dupa primele 3 etape de normalizare, relatia s-a transformat in urmatoarele tabele:

1. T3

id_recenzie	id_curs	nota	specificatii
4	5	7	ok
5	5	10	perfect
15	8	7	ok
16	8	6	-
18	8	8	bun
17	9	3	rau

2. T6

id_domeniu	denumire
2	IA
3	AG
7	BD

3. T7

id_curs	id_domeniu
5	2
5	7
8	2
9	3
9	7

4. T8

id_curs	id_creator	pret	link
5	12	14	[link 5]
8	9	12	[link 8]
9	3	6	[link 9]

5. T9:

id_creator	nume
12	Alex
9	Andrei
3	David

3.2 Exemplul 2

Pentru cel de-al doilea exemplu de normalizare la FN1 vom considera relatia reprezentata prin tabela T1:

id_creator	creator	id_domeniu	domeniu	id_curs	curs	data_crearii	pret
1	Alex	4, 5	AM, ALGAD	9	Curs1	19-8-2002	20
				10	Curs2	20-7-2009	25
2	Andrei	4, 6	AM, CN	11	Curs3	11-6-2003	30
3	David	7, 8	AG, SD	12	Curs4	29-5-2004	20

In baza de date se accepta ca 1 Creator sa creeze mai multe cursuri (relatie 1:N), 1 Creator sa fie specializat in mai multe domenii si 1 Domeniu sa fie asociat mai multor creatori (relatie M:N).

In tabela am adaugat atributul creator care reprezinta numele creatorilor, pentru a putea realiza normalizare in FN2.

3.2.1 Aducerea la FN1

Putem observa ca aceasta relatie nu se afla in FN1 deoarece exista valori multiple posibile in attributele id_curs, curs, data_crearii si pret. Pentru ca relatia sa contina doar valori atomice, vom imparti tabela T1 in 2 tabele: tabela T2 (va contine primele 4 atribute) si tabela T3 (va contine id_creator si ultimele 4 atribute).

Tabela T2:

id_creator	creator	id_domeniu	domeniu
1	Alex	4	AM
1	Alex	5	ALGAD
2	Andrei	4	AM
2	Andrei	6	CN
3	David	7	AG
3	David	8	SD

Tabela T3:

id_curs	id_creator	curs	data_crearii	pret
9	1	Curs1	19-8-2002	20
10	1	Curs2	20-7-2009	25
11	2	Curs3	11-6-2003	30
12	3	Curs4	29-5-2004	20

Vedem acum ca toate attributele contin doar valori atomice. Acest fapt ne spune ca relatia se afla in FN1.

3.2.2 Aducerea la FN2

In continuare vom aduce relatia T1 la FN2 utilizand tabelele T2 si T3. Este necesar sa cautam dependentele partiale de cheia primara.

In tabela T2 observam chiar 2 dependente partiale fata de cheia primara compusa (id_creator, id_domeniu). Atributul creator depinde doar de id_creator, in timp ce domeniu depinde doar de id_domeniu. Vom imparti tabela T2 in tabelele T4 (va contine id_domeniu si domeniu) si tabela T5 (va contine id_creator si id_domeniu) si T6 (id_creator si creator).

Tabela T4:

id_domeniu	domeniu
4	AM
5	ALGAD
6	CN
7	AG
8	SD

Tabela T5:

id_creator	id_domeniu
1	4
1	5
2	4
2	6
3	7
3	8

Tabela T6:

id_creator	creator
1	Alex
2	Andrei
3	David

În tabela T3 se observă că nu există dependente parțiale, deoarece cheia primară (id_curs) este singulară. Acest fapt se datorează relației dintre creator și curs, care are cardinalitate 1:N, înseamnă că fiecare curs are 1 singur creator. Astfel, cursul se poate identifica doar după id_curs și id_creator este cheia străină.

Nu mai există dependente parțiale față de cheia primară, ceea ce înseamnă că toate tabelele se află în acest moment în FN2.

3.2.3 Aducerea la FN3

Putem remarca în momentul acesta că tabelele T3, T4, T5 și T6 se află toate în FN3, așa că teoretic nu mai este nimic de făcut. Totuși, pentru a realiza și normalizare în FN3, putem modifica foarte puțin tabela T3 pentru a putea aplica și FN3 pe ea. Dacă adăugăm atributul creator în T3, atunci vom avea o dependență tranzitivă de la creator la id_creator la id_curs. Numim această tabela T3' deoarece vrem să putem diferenția între tabelele care au participat la proces de la început și cele care nu.

Tabela T3':

id_curs	id_creator	creator	curs	data_crearii	pret
9	1	Alex	Curs1	19-8-2002	20
10	1	Alex	Curs2	20-7-2009	25
11	2	Andrei	Curs3	11-6-2003	30
12	3	David	Curs4	29-5-2004	20

Putem scăpa de dependența tranzitivă prin împărțirea tabelului T3' în cele 2 tabele de mai jos. Tabela care conține id_creator și creator există deja, deci nu trebuie creată alta, ci vom folosi tot tabela T6.

Tabela T6:

id_creator	creator
1	Alex
2	Andrei
3	David

Restul atributelor sunt aceleasi cu cele din tabela T3, asa ca o putem folosi pe aceea.

Tabela T3:

id_curs	id_creator	curs	data_crearii	pret
9	1	Curs1	19-8-2002	20
10	1	Curs2	20-7-2009	25
11	2	Curs3	11-6-2003	30
12	3	Curs4	29-5-2004	20

Acum observam ca toate tabelele se afla in FN3, deci am terminat procesul de normalizare pe relatia curenta.

3.2.4 Rezultatul final

Relatia initiala:

id_creator	creator	id_domeniu	domeniu	id_curs	curs	data_crearii	pret
1	Alex	4, 5	AM, ALGAD	9	Curs1	19-8-2002	20
				10	Curs2	20-7-2009	25
2	Andrei	4, 6	AM, CN	11	Curs3	11-6-2003	30
3	David	7, 8	AG, SD	12	Curs4	29-5-2004	20

Dupa primele 3 etape de normalizare, relatia s-a transformat in urmatoarele tabele:

1. Tabela T3:

id_curs	id_creator	curs	data_crearii	pret
9	1	Curs1	19-8-2002	20
10	1	Curs2	20-7-2009	25
11	2	Curs3	11-6-2003	30
12	3	Curs4	29-5-2004	20

2. Tabela T4:

id_domeniu	domeniu
4	AM
5	ALGAD
6	CN
7	AG
8	SD

3. Tabela T5:

id_creator	id_domeniu
1	4
1	5
2	4
2	6
3	7
3	8

4. Tabela T6:

id_creator	creator
1	Alex
2	Andrei
3	David