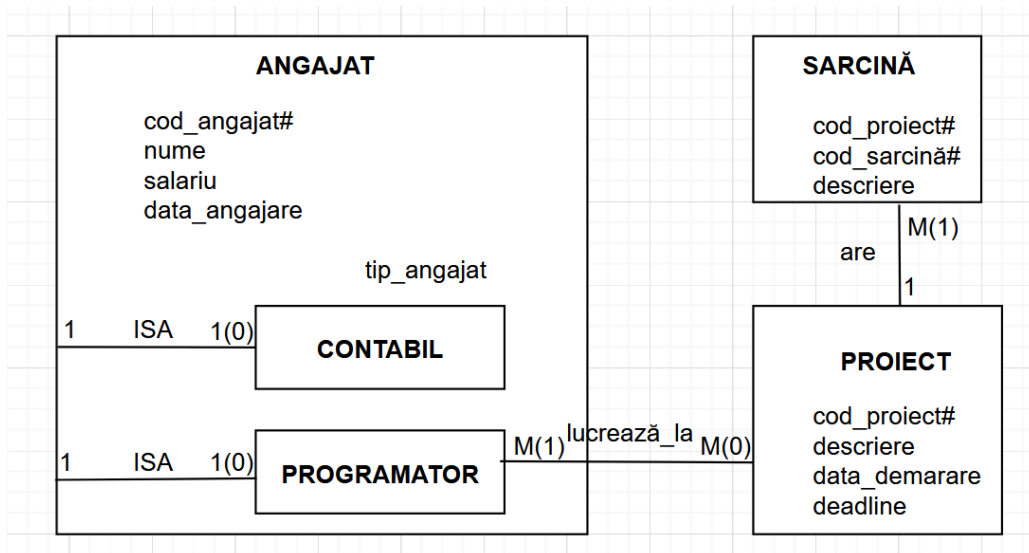


Tutoriat #2 Modelul Entitate-Relație (continuare)

Superentități și Subentități

- subentitatea este un "tip" de superentitate (este o submulțime a unei alte entități)
- definim subentități atunci când în model se precizează că pot exista mai multe "tipuri" dintr-un obiect, despre care ni se precizează și că **au proprietăți diferite**, esențiale pentru ceea ce modelăm, sau dacă ni se precizează că subentitățile **participă la relații diferite** (o subentitate are un rol distinct)
- între superentitate și fiecare subentitate a sa, se stabilește relația **ISA** (is a, "e un fel de")
- subentitatea va avea **aceeași cheie primară** ca și superentitatea
- atributele și relațiile superentității sunt valabile și pentru subentitate (**reciproca este falsă!**)
- o subentitate poate avea, la rândul ei, alte subentități

Exemplu: Într-o companie care prestează servicii software, există angajați de tip CONTABIL și PROGRAMATOR. Programatorii lucrează la proiecte software, care sunt alcătuite din sarcini.



Care este cheia primară a subentității CONTABIL? Poate un contabil să participe la un proiect? Dacă în PROGRAMATOR am adăuga atributul `limbaj_programare`, ar fi acesta "vizibil" în CONTABIL?

Observați că am definit relația *lucrează_la* doar între entitățile PROGRAMATOR și PROIECT.

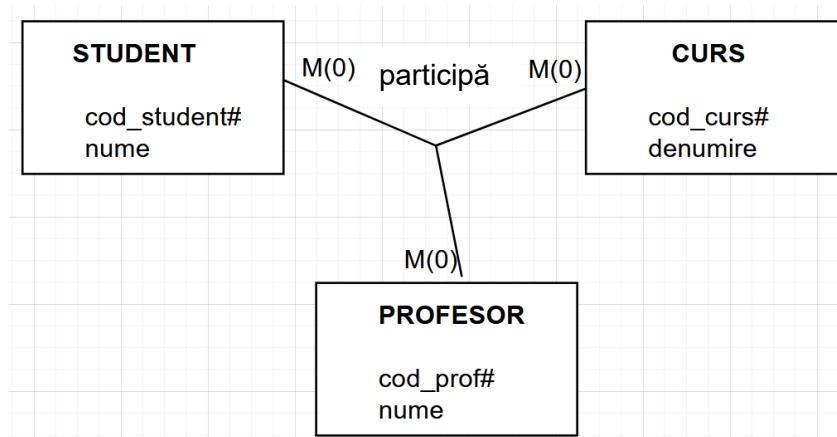
- În general, subentitățile sunt disjuncte (nu putem avea angajați care să fie și contabili, și programatori în același timp), însă depinde dacă din descrierea modelului se impune această constrângere sau nu.

Entități dependente

- În exemplul de mai sus, avem entitatea sarcină, ce desemnează un "task" al unui proiect. O sarcină nu poate exista de sine stătătoare, deoarece este determinată de proiectul căreia îi este asociată.
- SARCINĂ este o **entitate dependentă**.
- Cheia primară este compusă, include cheia primară a entității de care depinde + un atribut caracteristic.

Relații de tip 3

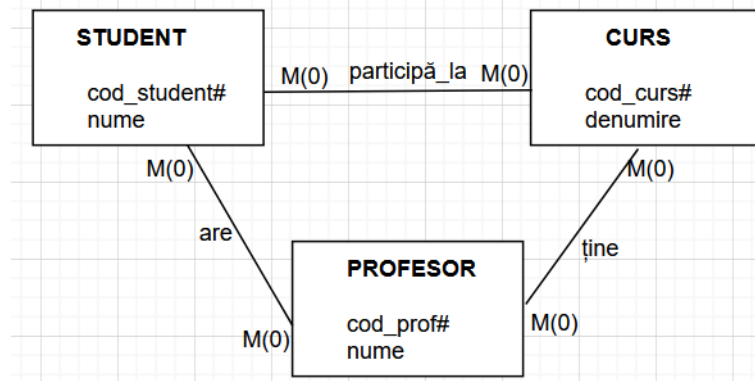
- O relație care implică trei entități, de exemplu: *Un student poate participa la mai multe cursuri, care sunt predate de către mai mulți profesori.*
- Ca să pot defini o relație de tip 3, trebuie să am "many" la fiecare capăt al acesteia (relație *many-to-many-to-many*)



Cum mă asigur că într-adevăr este vorba de o relație de tip 3? Verific punând întrebări:

- Cât studenți pot participa la un singur curs, predat de un singur profesor?
R: mai mulți -> pot avea (Ionel, Algebră, Prof. Stan) și (Costel, Algebră, Prof. Stan)
- Câte cursuri pot fi predate de un profesor, la care să participe (un) același student?
R: mai multe -> pot avea (Maria, Algebră, Prof. Ion) și (Maria, Analiză, Prof. Ion)
- Câți profesori pot fi titulari la un curs, și să predea aceluiși student?
R: mai mulți -> pot avea (Ana, Algoritmi, Prof. Popa) și (Ana, Algoritmi, Prof. Radu)

- O relație de tip 3 și 3 relații de tip 2 nu sunt echivalente! Prin spargerea unei relații de tip 3, pot rezulta informații inexacte.



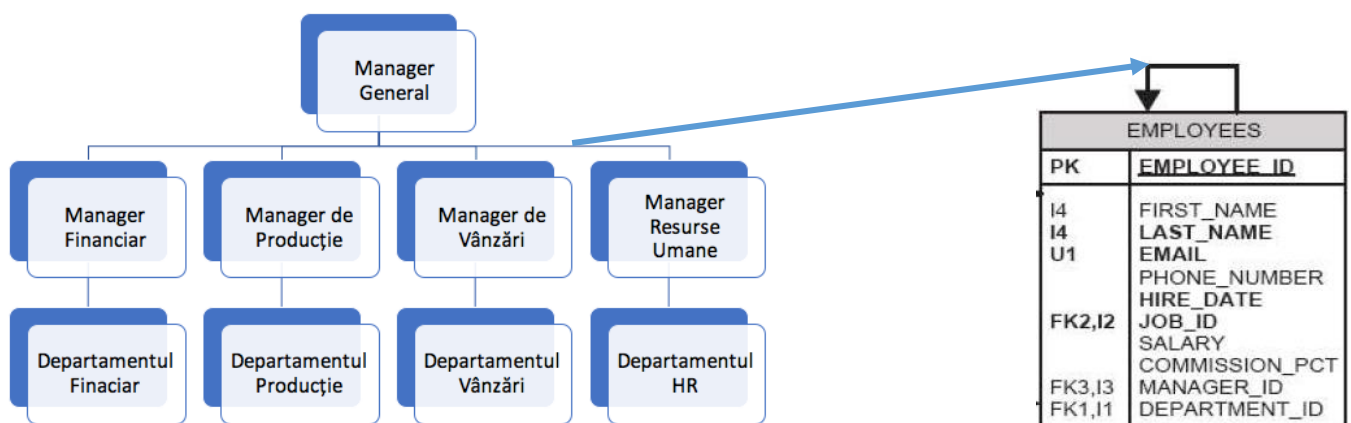
Explicație. Plecând de la descrierea modelului din exemplul de mai sus, presupunem că studentul **St1** s-a înscris la cursul **BD** când acesta era predat de către **Prof1**. Mai presupunem și că **BD** a fost predat atât de **Prof1**, cât și de **Prof2**. Cele trei relații binare ar memora următoarele asocieri:

- Student-Curs: (**St1**, **BD**)
- Curs-Profesor: (**BD**, **Prof1**), (**BD**, **Prof2**)
- Profesor-Student: (**Prof1**, **St1**)

Dacă încercăm să recompunem informațiile din aceste relații binare (de exemplu, un join între Student-Curs și Curs-Profesor), am putea obține, în plus, informația (eronată) că **St1** este înscris la cursul de **BD**, care este predat de către **Prof2**, deși inițial aveam doar asocierea (St1, BD, Prof1).

Relații recursive ("de ordinul 1")

- Dacă avem o ierarhie a instanțelor dintr-o entitate, se pot defini relații recursive (de la o entitate la ea însăși)



Regula modelului: *Un angajat poate avea în subordinea sa mai mulți angajați*

Exerciții

Proiectați diagrama E/R, pornind de la următoarele descrieri ale modelelor și reguli de funcționare:

1. Modelul de date va gestiona informații legate de rezervarea și organizarea activităților din cadrul unei agenții de turism. Într-o agenție de turism, lucrează ghizi, șoferi și agenți de vânzări. Agenția se ocupă de clienții care doresc să achiziționeze sejururi sau excursii. Clienții sunt preluați de agenții de vânzări, după care, în urma ofertelor, ei aleg un sejur sau o excursie din oferta agenției. Ghizii conduc excursii, la care pot fi atașați mai mulți șoferi.

Reguli de funcționare:

- Din oferta agenției fac parte doar sejururi și excursii.
- O excursie poate avea un singur ghid, dar turiștii pot opta pentru excursii fără ghid
- Pot exista ghizi care să nu fie alocați excursiilor.
- O excursie poate avea mai mulți șoferi. Nu este obligatoriu ca orice ofertă a agenției să aibă un șofer!
- Clienții pot fi deserviți de către mai mulți agenți de vânzări și pot alege mai multe oferte.
- Un sejur se desfășoară într-o anumită locație, iar într-o locație au loc mai multe sejururi. Pot exista și locații în care să nu aibă loc sejururi.

2. O companie de telecomunicații oferă servicii variate (telefonie, Internet, TV) abonaților săi. Abonamentele pot fi pentru servicii individuale sau pentru pachete de servicii aflate la un moment dat în oferta companiei. Abonații pot raporta incidente (defecțiuni) în legătură cu un serviciu pe care îl dețin. Incidentele sunt preluate de angajați de tip operator și atribuite unei echipe de intervenție ce conține angajați de tip tehnician. La final, pentru fiecare intervenție se va raporta modul de soluționare și durata intervenției (de către un membru al echipei).

Reguli de funcționare:

- Mai mulți abonați pot avea același tip de abonament
- Un tehnician poate face parte dintr-o singură echipă de intervenție
- O echipă de intervenție poate soluționa o singură intervenție
- Un abonat poate raporta mai multe incidente, iar fiecare incident este raportat de către un singur abonat.

3. Studenții unei facultăți participă la stagii oferite de către companii. Stagiile pot fi de tip dezvoltare sau suport_tehnic. Fiecare stagiou are o dată de început și o dată de final. Un stagiou de tip dezvoltare necesită cunoașterea unor limbaje de programare. Pe perioada efectuării unui stagiou, studenții pot fi coordonați de mai mulți specialiști din compania respectivă. În timpul unui stagiou de tip dezvoltare, un student poate fi încadrat pe mai multe proiecte. În cadrul unui proiect, studentul poate realiza task-uri specifice, iar în fiecare dintre aceste task-uri, poate lucra cu mai multe tehnologii (limbaje, sisteme, etc.)