**CURS 2**

**Modelare entitate - relatie**

* Model = reprezentare a obiectelor şi evenimentelor lumii reale şi a asocierilor dintre ele.
* Abstractizare asupra aspectelor semnificative ale unei „întreprinderi“, ale unui sistem real
* Caz particular al deosebirii uzuale dintre logic și fizic.
* 3 tipuri de modele (care descriu aspecte … ale procesului modelat):
  + statice
  + dinamice
  + functionale
* Model de date = o colecţie integrată de concepte necesare descrierii:
  + datelor
  + relatiilor dintre ele
  + constrângerilor existente asupra datelor sistemului real analizat
* Modelarea unei baze de date = trecerea de la percepţia unor fapte din lumea reală la reprezentarea lor prin date
* Modelul de date trebuie:
  + să reflectefidel fenomene ale lumii reale
  + să urmărească evoluţiaacestei lumi
  + să permită comunicarea dintre fenomenele lumii reale
* Modelul de date are 3 componente:
  + o mulţime de reguli conform cărora sunt construite bazele de date (partea structurală)
  + o mulţime de operaţii permise asupra datelor, care sunt utilizate pentru reactualizarea sau regăsirea datelor (partea de prelucrare)
  + o mulţime de reguli de integritate, care asigură coerenţa datelor
* Modelarea semantica a datelor are 4 etape:
  + identificarea unei multimi de comcepte semantice utile descrierii lumii reale
    - Se presupune că lumea reală (modelul real analizat) este formată din entităţi care fiecare are anumite proprietăţi (identitate, că există legături, corelaţii între entităţi).
  + cautarea unei mulţimi de obiecte formale, simbolice, utilizate pentru reprezentarea conceptelor semantice anterioare
  + se dau reguli de integritate formale şi generale (constrângeri) care să reflecte restricţiile la care este supus modelul
  + definirea unei mulţimi de operatori formali prin care pot fi prelucrate şi analizate obiectele formale.

**Modelul Entitate-Relatie**

* abordare a modelarii semantice
* model de date conceptual, pt a usura proiectarea BD
* de nivel inalt, independent de platforma hardware utilizata si de tipul SGBD-ului
* reprezentat grafic prin diagrame E/R
* modelul E/R imparte elementele unui sistem real in 2 categorii:
  + entitati
  + relatii (legaturi, asocieri) intre entitati
* entitatile si legaturile au caracteritici (atribute)
* conceptul de relaţie, în sensul de asociere, care intervine în definirea diagramei E/R ≠ conceptul de relaţiecare este specific modelului relaţional

**Diagrame entitate - relatie**

* Diagrama E/R = model de date conceptual de nivel inalt neformalizat pentru reprezentarea unui sistem din lumea reală
* Entitate = persoană, loc, concept, activitate, eveniment care este semnificativ pentru ceea ce modelăm
* Entităţile:
  + vor deveni tabele în modelul relaţional
  + scrise cu litere mari
  + substrantive
  + obligatoriu o descriere detaliata
  + in aceeasi diagrama nu pot exista 2 entitati cu acelasi nume sau aceeasi entitate cu nume diferite
* Cheia primară/PK = un identificator unic în cadrul entităţii, făcând distincţie între valori diferite ale acesteia
* Cheia primara:
  + unica
  + cunoscuta in orice moment
  + fara inf. discriptive
  + simpla, fara ambiguitati
  + stabila
  + controloata de administratorul BD
  + familiara utilizatorului
* Relaţie = comunicare între 2 sau mai multe entităţi
* Existenţa unei relaţii este subordonată existenţei entităţilor pe care le leagă
* in modelul relational, relatiile devin tabele speciale sau coloane speciale care refera chei primare
* relatiile sunt verbe
* fiecare relatie trebuie sa aiba o descriere detaliata
* aceeasi diagrama poate avea relatii diferite cu acelasi nume
* fiecare relatie are o cardinalitate (max/min) = nr de tupluri ce aparint relatiei
* POATE (cardinalitate maxima) → TREBUIE (cardinalitate minima):
  + Cati salariati **pot** lucra intr-un depart? Multi
  + In cate depart **poate** lucra un salariat? In cel mult unul
    - relatia este many-one (n:1)
  + Cati salariati **trebuie** sa conduca un depart? Cel putin unul
  + Cate depart **trebuie** sa conduca un salariat? Zero
    - relatia este one-zero (1:0)
* Atribut = proprietate descriptiva a unei entitati sau a unei relatii
* Atributul:
  + devine coloana in modelele relationale
  + valoarea lui devine valoare in coloane
  + substantive
  + ii trebuie o descriere completa: ex, contra-ex, caracteristici
  + fiecare atribut are specificat:
    - numele
    - tipul fizic (integer, float, char etc.)
    - valori posibile
    - valori implicite
    - reguli de validare
* Reguli (neunice) pentru poriectarea unei diagrame E/R:
  + entitatile → reprezentate prin dreptunghiuri
  + relatiile → reprezentate prin arce neorientate
  + atributele chei primare → subliniate sau marcate prin “#”, plasat la sf numelui acestora
  + card min → indicata de paranteze
  + card max → fara paranteze
  + nu e necesar specificate toate atributele in diagrama
* Cazuri speciale de entitati, relatii, atribute:
  + dependenta
    - Entitate dependenta → nu poate exista in mod independent
    - Cheia primara a unei entitati dependente include cheia primara a sursei (nr\_proiect) si cel putin de un atribut caracteristic entitatii (nr\_sarcina)
    - Entitatea dependenta se deseneaza prin dreptunghi cu linii mai subtiri
  + mostenirea atributelor
    - Subentitatea/subclasa = submultime a unei entitati (superentitate/superclasa)
      * ex: Salariat ↔ programator
    - Subentitatea este desenata prin dreptunghiuri incluse in superentitate
    - ISA = relatia subentitate-superentitate cu card max 1:1 si card min 1:0
    - Cheile primare, atributele si relatiile unei superentitati sunt valabile pentru orice subentitate (reciproca e falsa)
  + generalizare
    - din entitati similare cu mai multe atribute comune se pot creea superentitati
    - superentitatile contin atribute comune, iar atributele speciale sunt asignate la subentitati
    - pt noile superentitati se introduc chei primare artificiale
  + specializare
    - dupa valorile unor atribute clasificatoare se pot determina clase
    - un grup de subentitati reciproc exclusive defineste o clasa
    - clasele se aliniaza in desen vertical
  + relatii recursive
  + relatii de tip 2,3
    - relatii relative la 2 entitati → de tip 2
    - relatii relative la 2+ entitati → de tip 3
    - 3 relatii de tip 2 ≠ o relatie de tip 3
    - rupand o relatie de tip 3 in 3 relatii de tip 2 → informatii incorecte
  + relatie sau atribut
    - dacă un atribut al unei entităţi reprezintă cheia primară a unei alte entităţi, atunci el referă o relaţie
      * ex: cod\_departament în tabelul SALARIAT
  + entitate sau relatie
    - se cercetează cheia primară
    - dacă PK combină cheile primare a două entităţi, atunci este vorba de o relaţie
      * ex: cheia primară a relaţiei asociat\_la combină cod\_salariat cu nr\_proiect, prin urmare, SALARIAT\_asociat la\_PROIECT va defini o relaţie şi nu o entitate
* Relatiile indirecte → excluse din model pt ca pot conduce la redundanta in BD
* Atributele derivate → eliminate si introduse expresii prin care acestea pot fi calculate
* Algortimul pt proiectarea diagramei E/R:
  + identificare entitatilor din cadrul sistemului analizat
  + identificarea relatiilor dintre dintre entitati si stabilirea cardinalitatii
  + identificarea atributelor aferente entitatilor si asocierilor dintre ele
  + stabilirea atributelor de indentificare a entitatilor → stabilirea PKs
* Aceeaşi realitate poate fi percepută diferit de către diverşi analişti pentru un acelaşi sistem → modele structurale distincte
* Pasii urmatori ⇒ model relational care va elimina anumite clase anomalii ce pot aparea in proiectarea modelului de date
  + construirea diagramei conceptuale
  + obtinerea schemelor relationale
  + normalizarea acestora

**Modelul E/R extins**

* Conceptele de bază ale modelării E/R nu sunt suficiente pentru a reprezenta cerinţe complexe
* Modelul E/R susţinut cu concepte semantice adiţionale defineşte modelul E/R extins (EER)
  + include toate conceptele modelului original
    - conceptele adiţionale de subclasă, superclasă, moştenire, specializare, generalizare
* Superclasa/superentitatea = o entitate care include subclase/subentităţi distincte, ce trebuie reprezentate în modelul de date
* Subclasa are un rol distinct şi, evident, este membră a unei superclase
* O subclasă, fiind o entitate, poate să posede propriile subclase
* O entitate împreună cu subclasele ei, subclasele acestora şi aşa mai departe defineşte o ierarhie de tip (ierarhie de specializare).
  + ex: ANGAJAT\_TEMP reprezintă o superclasă pentru entitatea MODEL
* Specializarea = procesul de maximizare a diferenţelor dintre membrii unei entităţi, prin identificarea caracteristicilor distinctive ale acestora
  + constrângere de disjuncţie = dacă subclasele unei specializări sunt disjuncte, atunci o entitate poate fi membră doar a unei subclase a acesteia (constrângere de disjuncţie).
  + constrângere de participare = specializare cu participare totală → fiecare entitate din superclasă trebuie să fie membră a unei subclase din specializare
  + o specializare cu participare parţială → nu este necesar ca o entitate să aparţină vreunei subclase a acesteia.
    - ex: există salariaţi în PERS\_CONTACT care nu aparţin niciunei subentităţi ale acesteia
* Generalizarea = procesul de minimizare a diferenţelor dintre entităţi, prin identificarea caracteristicilor comuneale acestora
* Generalizarea are ca rezultat identificarea unei superclase generalizate din subclasele iniţiale

**Deficiente ale modelului E/R**

* Cauza: interpretare eronata a sensului unei relaţii → capcane de conectare
* Posibil sa necesite restructurarea modelului
* 2 clase de capcane de conectare:
  + de intrerupere
  + in evantai
* Capcane de întrerupere = modelul sugerează existenţa unei relaţii între entităţi, dar nu există o cale între anumite apariţii ale entităţilor.
  + apare acolo unde există o relaţie cu participare parţială (card min=0), care face parte din calea dintre entităţile ce sunt legate
* Capcane în evantai = modelul ia în considerare o relaţie între entităţi, dar calea dintre anumite apariţii ale entităţilor este ambiguă
  + apar când două sau mai multe relaţii one-to-many provin din aceeaşi entitate
* Aceste capcane generează situaţiile în care, aşa cum a fost proiectat modelul de date, el nu poate să răspundă la anumite interogări
* Ex capcana de intrerupere: pentru a afla pentru ce prezentare de modă a fost creată o anumită vestimentaţie, a fost necesară introducerea unei legături între entităţile PREZENTARE şi VESTIMENTATIE, care însă a generat redundanţă în modelul de date

**Exemple de diagrame E/R**

* Exemplu 1 → gestiunea activitatilor de editare dintr-o editura
* Exemplul 2 → gestiunea unei firme de construcutii
* Exemplul 3 → descoperiri de monede antice din România
  + STANŢA (nr\_stanţă, împărat emitent, valoare nominală, an emitere, monetăria, legenda de pe avers, legenda de pe revers) → atribute ale entităţii STANTA
* Exemplul 4 → evidența școlilor de șoferi din România
  + Completaţi relaţiile (lucreaza\_la, conduce, sustine, asista, instruieste) dintre entităţi şi specificaţi cardinalitatea