

4. IMPLEMENTAREA MODELULUI CONCEPTUAL

- Definirea modelelor de baze de date
- Caracteristicile principale ale bazelor de date relaționale
- Transformarea modelului conceptual al bazei de date (ERD) în modelul fizic (tabelele bazei de date)
- Principalele operații care se pot efectua asupra bazelor de date
- Rolul regulilor de integritate și care sunt acestea
- Programe de validare și acțiuni

4.1. MODELE DE BAZE DE DATE

- **Bazele de date (B.D.)** au fost concepute pentru a stoca volume mari de informații omogene de date între care se pot stabili relații.
- Baza de date este o colecție structurată de date aflate în interdependență, ce pot fi consultate pentru a răspunde diferitelor interogări. Înregistrările returnate în urma interogărilor devin **informații**.



- Sistemul de programe ce permite **descrierea, organizarea, memorarea, regăsirea, administrarea și securizarea** informațiilor dintr-o bază de date se numește **sistem de gestiune a bazelor de date (S.G.B.D.)**. El este un software special asociat bazelor de date ce asigură interfața între o bază de date și utilizatorii ei, rezolvând cererile de acces la datele memorate.
- Pentru orice bază de date poate fi dată o descriere a datelor și obiectelor memorate și relațiile existente între ele. Această descriere se numește **schemă**.




- Un **model de bază de date** definește un set de operații care pot fi realizate cu datele respective.
- Cele mai cunoscute modele de baze de date sunt:
 - **modelul tabelar** – datele sunt memorate sub forma unui tabel bidimensional de date;
 - **modelul ierarhic** – datele sunt organizate sub forma unor structuri arborescente
 - **modelul rețea** – este o colecție de noduri și legături fiecare nod putând fi legat de oricare altul
 - **modelul relațional** – datele sunt organizate sub formă de tabele între care există diverse legături
 - **modelul obiectual** – destinat să suporte modele de obiecte complexe, este asemănător rețelei și modelului ierarhic
 - **modele hibride** – sunt mixturi ale modelelor prezentate anterior.



4.2. BAZE DE DATE RELAȚIONALE

- Au la bază teoria matematică a relațiilor ceea ce face posibilă tratarea algoritmică a proiectării bazelor de date și problema normalizării.
- Modelul relațional este bazat pe algebra relațională și limbajele de programare sunt sub forma unui software specializat ce asistă procesul de implementare a bazelor de date. Ele sunt **SQL** (Structured Query Language) și **QBE** (Query By Example).
- **Oracle** este un sistem de gestiune a bazelor de date ce are la bază tehnologia **client – server**.



- Transformarea modelului conceptual, a ERD-ului în model fizic adică în baza de date propriu-zisă se numește **mapare**. Procesul implică transformarea fiecărui element al ERD-ului.
 - Prima etapă este crearea tabelii bazei de date.
 - fiecărei **entități** îi corespunde câte un **tabel**. Numele lui este un *substantiv la plural*. Entitatea ANGAJAT se transformă în tabela ANGAJAȚI.
 - fiecare **atribut** al entității devine o **coloană** a tabelii. Fiecare coloană a tabelii va memora date de același tip.
 - fiecare **instanță** a unei entități se va transforma într-un **rând** (**înregistrare** a tabelului)
 - identificatorul unic al entității devine cheie primară a tabelii. Coloana sau combinația de coloane care identifică în mod unic toate liniile unui tabel se numește **cheie primară**.
- 

- Orice tabelă are linii și coloane și conține date organizate conform unei structuri. În limbajul bazelor de date, coloanele se numesc **câmpuri**. Fiecare coloană reprezintă un câmp cu o denumire unică, de un anumit tip (șir de caractere, numeric, dată calendaristică) ce are o dimensiune precisă.
- Rândurile tabelului se numesc **înregistrări**.
- Informațiile despre o tabelă a bazei de date sunt prezentate în **diagramele de tabele** care sunt niște tabele în care se notează numele coloanelor pe care le va avea tabela, tipul datelor memorate în coloane, dacă o coloană este cheie primară se notează **pk** în dreptul ei, cheia străină se notează **fk** și în ultima coloană se notează dacă atributul este opțional sau obligatoriu.



Numele coloanei	Tip	Tip cheie	Opționalitate
titlu	varchar2	pk	*
autor	varchar2	pk	*
data_apariției	date		*
format	varchar2		*
nr_pagini	number		*



Tipul de date	Descriere	Dimensiune Maximă
VARCHAR2	Șir de caractere de lungime variabilă	4000 bytes
CHAR	Șir de caractere de lungime fixă	2000 bytes
NUMBER(p,s)	Număr având p cifre din care s la partea zecimală. (s negativ reprezintă numărul de cifre semnificative din fața punctului zecimal)	p (precizia) între 1 și 38. s (scala) între -84 și 127.
DATE	Data calendaristică	De la 1 Ianuarie 4712 BC pana la 31 Decembrie, 9999 AD.
TIMESTAMP	Se memorează data calendaristică, ora, minutul, secunda și fracțiunea de secundă	Fracțiunea de secundă este memorată cu o precizie de la 0 la 9.
INTERVAL YEAR TO MONTH	perioadă de timp în ani și luni.	
INTERVAL DAY TO SECOND	memorează un interval de timp în zile, ore, minute și secunde	
CLOB	Character Large Object	4 Gigabytes
BLOB	Binary Large Object	4 Gigabytes
BFILE	Se memorează adresa unui fișier binar de pe disc	4 Gigabytes



ELEV

- # cnp
- * nume
- * data_nasterii
- * adresa
- o telefon
- o email

ELEVI

Numele coloanei	Tip	Tip cheie	Opționalitatea
cnp	varchar2	Pk	*
nume	varchar2		*
prenume	varchar2		*
data_nasterii	date		*
adresa	varchar2		*
telefon	varchar2		o
email	varchar2		o

CURS

Numele coloanei	Tip	Tip cheie	Opționalitatea
id	number	Pk	*
denumire	varchar2		*
durata	interval year to month		*
nr_credite	number		*

CURS

- # id
- * denumire
- * durata
- * nr_credite

MASINI

Numele coloanei	Tip	Tip cheie	Opționalitatea
serie_motor	varchar2	Pk	*
nr_inmatriculare	varchar2		*
model	varchar2		*
culoare	varchar2		*
capacitate_cilindrică	number		*
consum	number		o
emisie_noxe	number		o

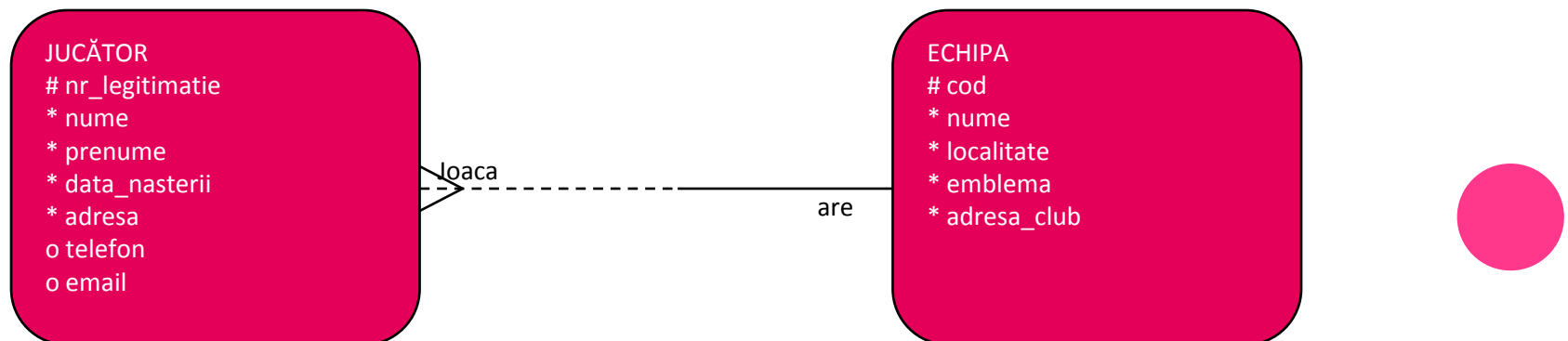
MAȘINĂ

- # serie motor
- * nr_inmatriculare
- * model
- * culoare
- * capacitate_cilindrică
- * consum

4.3. MAPAREA RELAȚIILOR

Maparea relațiilor 1:M

- Diagrama se citește astfel:
- Fiecare JUCĂTOR poate juca la o ECHIPĂ și numai una.
- La fiecare ECHIPĂ trebuie să joace unul sau mai mulți JUCĂTORI.
- Nu se pot memora toți jucătorii care joacă la o echipă în cadrul tabelii ECHIPA deoarece o coloană ar avea valori multiple. Invers, se poate memora pentru fiecare jucător, echipa la care joacă deoarece acesta nu poate juca decât la o singură echipă. Pentru fiecare jucător se memorează codul echipei la care joacă.



JUCĂTORI

Numele coloanei	Tip	Tip cheie	Opționalitatea
nr_legitimatie	number	Pk	*
nume	varchar2		*
prenume	varchar2		*
data_nasterii	date		*
adresa	varchar2		*
telefon	number		0
email	varchar2		0
cod echipă	number	fk	0

Dacă relația pe partea **many** este opțională atunci și coloanele cheii străine vor fi opționale. Un jucător poate la un moment dat să nu joace la nici o echipă caz în care câmpul `cod_echipă` va rămâne necompletat în dreptul lui (va avea valoarea NULL). Dacă relația este obligatorie pe partea **many**, atunci coloanele care fac parte din cheia străină vor fi obligatorii.

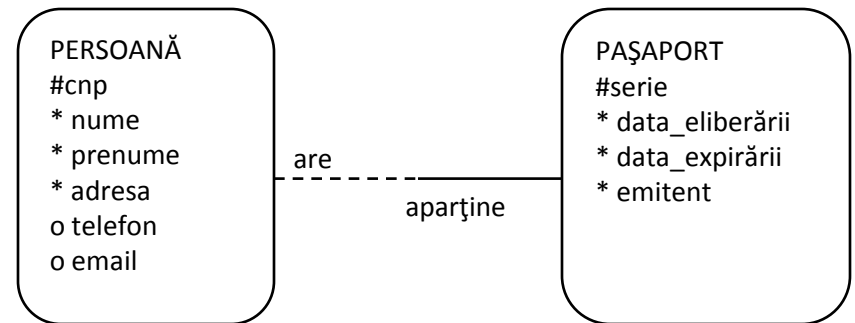


- În general, la maparea unei relații de tip many to many vom introduce în tabela corespunzătoare entității de pe partea many a relației cheia primară a entității de pe partea one a relației. Câmpurile astfel introduse se vor numi cheie străină (fk)
- Reguli:
 - **cheia străină** a unei tabele este cheia primară a tablei referință
 - **cheia străină** este întotdeauna introdusă în tabela corespunzătoare entității din partea many a relației.



4.4. MAPAREA RELAȚIILOR 1:1

- Pentru două entități A și B care se află în relație 1:1, maparea se realizează incluzând cheia primară a entității A în cadrul tabelii B dar se poate foarte bine și invers, deoarece fiecărei instanțe a entității A îi corespunde o instanță a entității B.
- Pentru fiecare persoană se poate memora seria de la pașaport dar și invers, pentru fiecare pașaport putem memora cnp-ul deținătorului.



Persoane

Numele coloanei	Tip	Tip cheie	Opționalitatea
cnp	number	Pk	*
nume	varchar2		*
prenume	varchar2		*
adresa	varchar2		*
telefon	number		0
email	varchar2		0
serie	number	fk	0

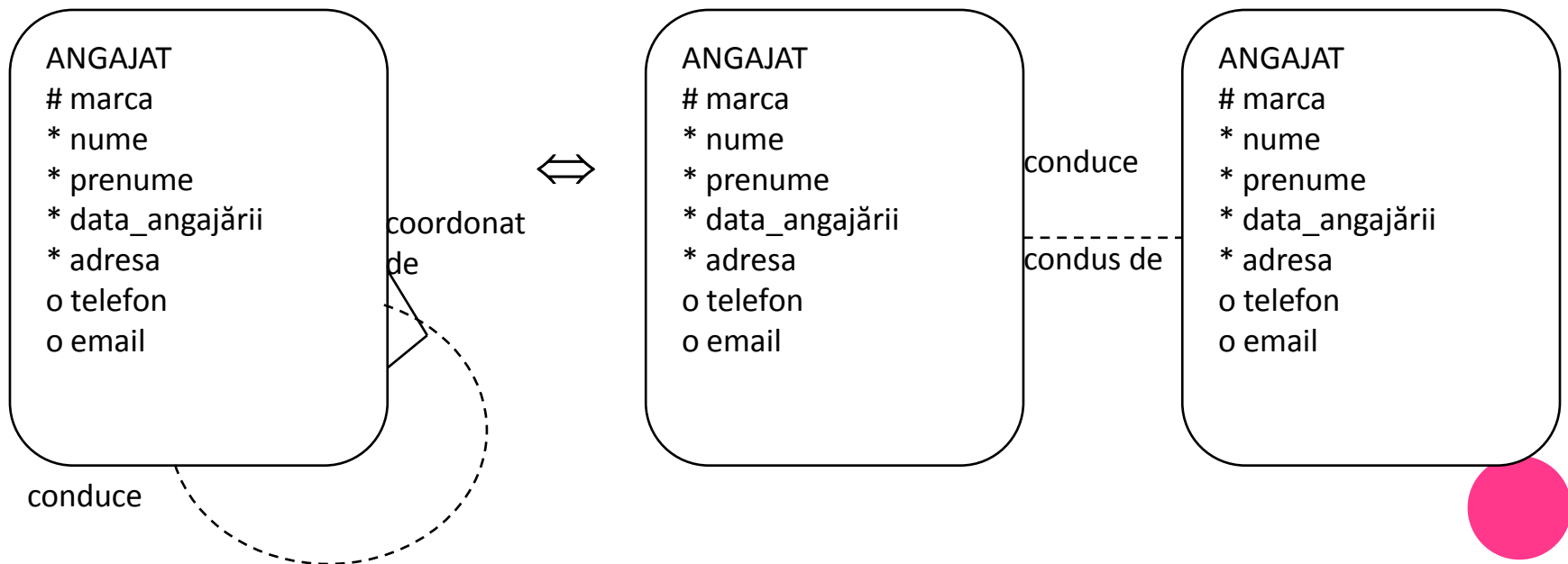
Pașapoarte

Numele coloanei	Tip	Tip cheie	Opționalitatea
serie	number	pk	*
data_eliberarii	date		*
data_expirarii	date		*
emitent	varchar2		*
cnp	Number	Fk	*

Decizia depinde de specificul afacerii modelate. Dacă ne interesează de exemplu în primul rând persoanele și apoi datele de pe pașaport, atunci vom adopta prima variantă, memorând seria de pașaport în cadrul tabelii PERSOANE, dacă însă baza de date este destinată evidenței pașapoartelor, atunci se va adopta a doua variantă.

4.5. MAPAREA RELAȚIILOR RECURSIVE

- O relație recursivă poate fi privită ca o relație de tipul 1:M între o entitate și ea însăși.



- Tablela ANGAJAȚI va arăta astfel:

ANGAJAȚI

Numele coloanei	Tip	Tip cheie	Opționalitatea
marca	number	Pk	*
nume	varchar2		*
prenume	varchar2		*
data_angajării	date		*
adresa	varchar2		*
telefon	varchar2		0
email	varchar2		0
marca_sef	number	fk	0



4.6. MAPAREA RELAȚIILOR BARATE

- Relațiile barate se transformă în urma mapării în chei străine în tabela aflată în partea many a relației, la fel ca la maparea unei relații 1:M. Bara de pe relație exprimă faptul că acele coloane ce fac parte din cheia străină vor deveni parte a cheii primare a tablei din partea many a relației barate.
- Exemplu: Cheia primară a tablei **ATTRIBUTE** va fi formată din colanele `denumire_atribut` și `denumire_entitate`, aceasta din urmă fiind de fapt cheie străină în tabela **ATTRIBUTE**:

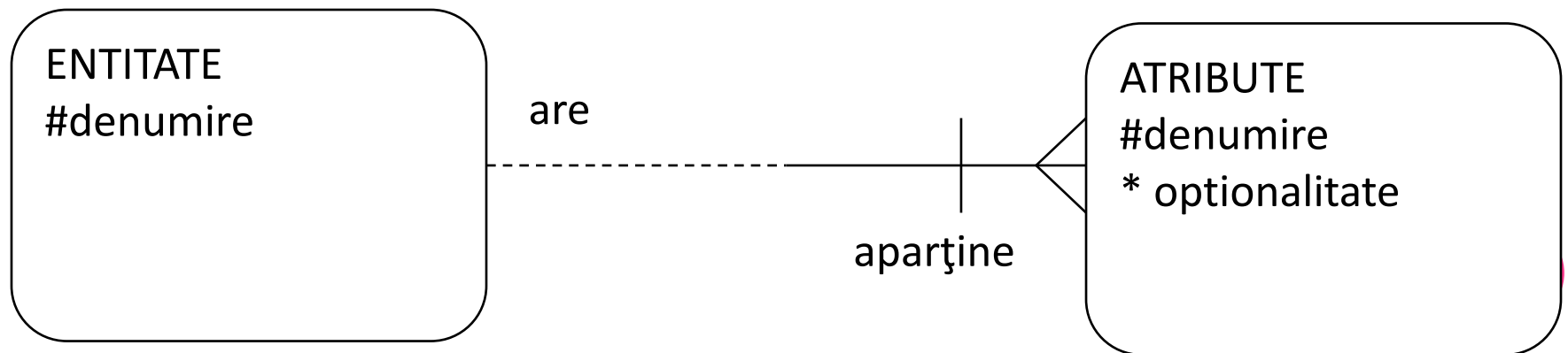


Tabela ENTITĂȚI

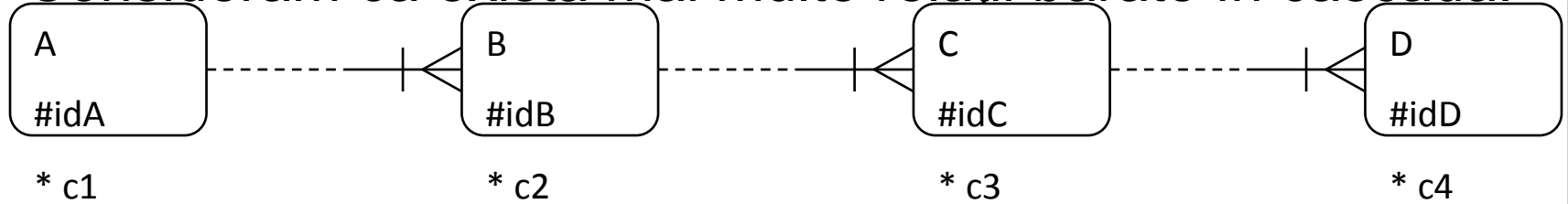
Numele coloanei	Tip	Tip cheie	Opționalitatea
denumire	varchar2	pk	*

Tabela ATRIBUTE

Numele coloanei	Tip	Tip cheie	Opționalitatea
denumire_atribut	varchar2	pk	*
denumire_entitate	varchar2	pk, fk	*
opționalitate	varchar2		*



- Considerăm că există mai multe relații barate în cascadă:



Numele coloanei	Tip cheie	Opționalitatea
idA	pk	*
c1		*

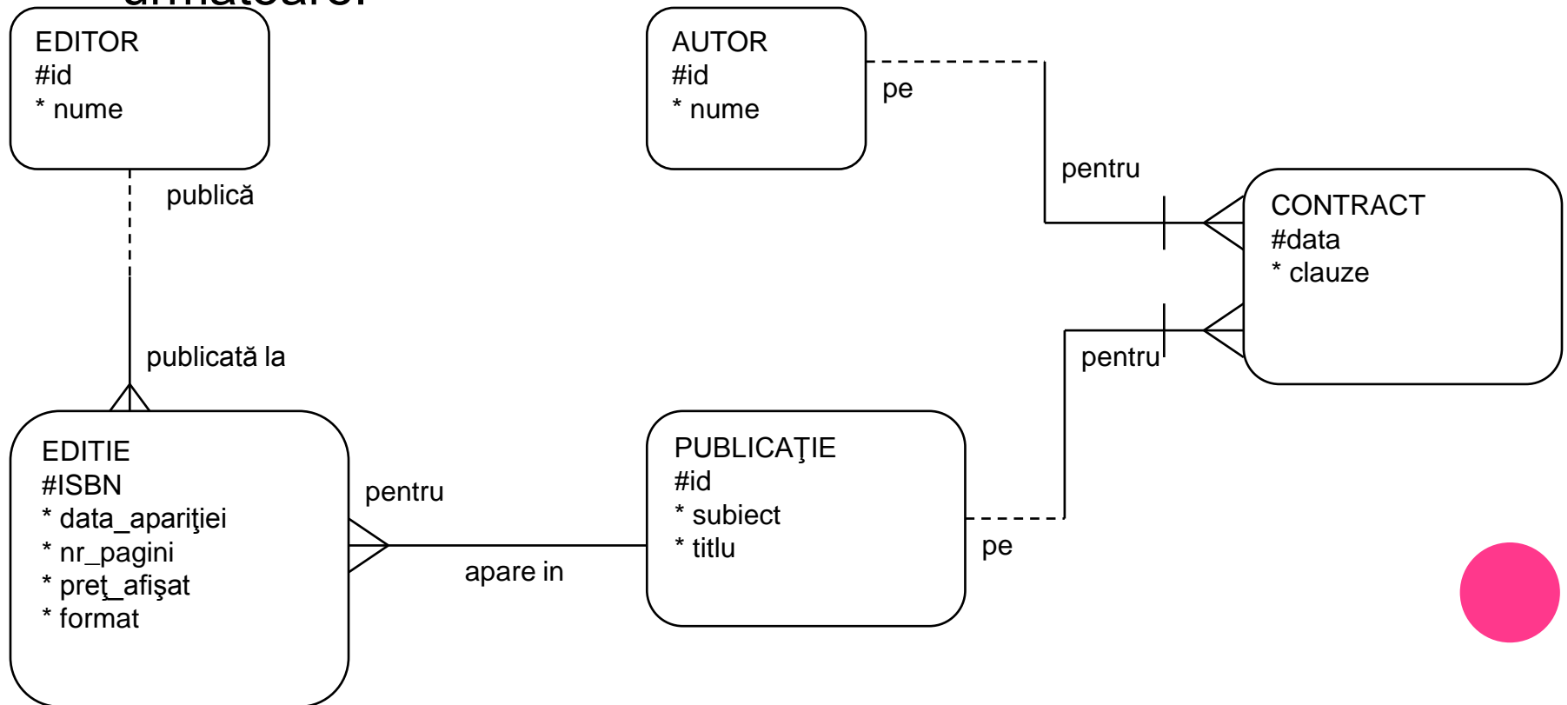
Numele coloanei	Tip cheie	Opționalitatea
idB	pk	*
c2		*
idA	pk, fk	*

Numele coloanei	Tip cheie	Opționalitatea
idC	pk	*
C3		*
idA	pk, fk	*
idB	pk, fk	*

Numele coloanei	Tip cheie	Opționalitatea
idD	pk	*
c4		*
idA	fk	*

4.7. EXEMPLU COMPLET DE MAPARE

- Se consideră operația de mapare a ERD-ului din figura următoare:



- Diagramele de tabele corespunzătoare fiecărei entități sunt:

Tabela EDITORI

Numele coloanei	Tip	Tip cheie	Opționalitatea
id	number	pk	*
nume	varchar2		*

Tabela EDIȚII

Numele coloanei	Tip	Tip cheie	Opționalitatea
ISBN	number	pk	*
data apariției	date		*
nr_pagini	number		*
preț afișat	number		*
format	varchar2		*
id_editor	number	fk1	*
id_publicatie	number	fk2	*

Tabela PUBLICAȚII

Numele coloanei	Tip	Tip cheie	Opționalitatea
id	number	pk	*
subiect	varchar2		*
titlu	varchar2		*

Tabela AUTORI

Numele coloanei	Tip	Tip cheie	Opționalitatea
id	number	pk	*
nume	varchar2		*

Tabela CONTRACTE

Numele coloanei	Tip	Tip cheie	Opționalitatea
data	date	pk	*
clauze	varchar2		*
id_autor	number	pk, fk	*
id_publicatie	number	pk, fk	*

În ultima diagrama, în cazul relațiilor barate, cheia străină face parte din cheia primară a entității respective.



4.8. OPERAȚII SPECIFICE PRELUCRĂRII BAZELOR DE DATE

Orice sistem de gestiune a bazelor de date trebuie să asigure următoarele funcții:

- definirea structurii bazei de date
- încărcarea datelor în baza de date (adăugarea de noi înregistrări în baza de date)
- accesul la date pentru:
 - interogare
 - ștergere
 - modificare
- întreținerea bazei de date:
 - refacerea bazei de date prin existența unor copii de siguranță
 - repararea în caz de incident
 - colectarea și refolosirea spațiilor goale
- posibilitatea de reorganizare a bazei de date prin:
 - restructurarea datelor
 - modificarea accesului la date
- securitatea datelor

O parte a acestor operații se pot realiza cu ajutorul limbajului SQL, altele cu ajutorul unor programe specializate care sunt puse la dispoziția administratorului bazei de date de către sistemul de gestiune a bazelor de date.



4.9. REGULI DE INTEGRITATE

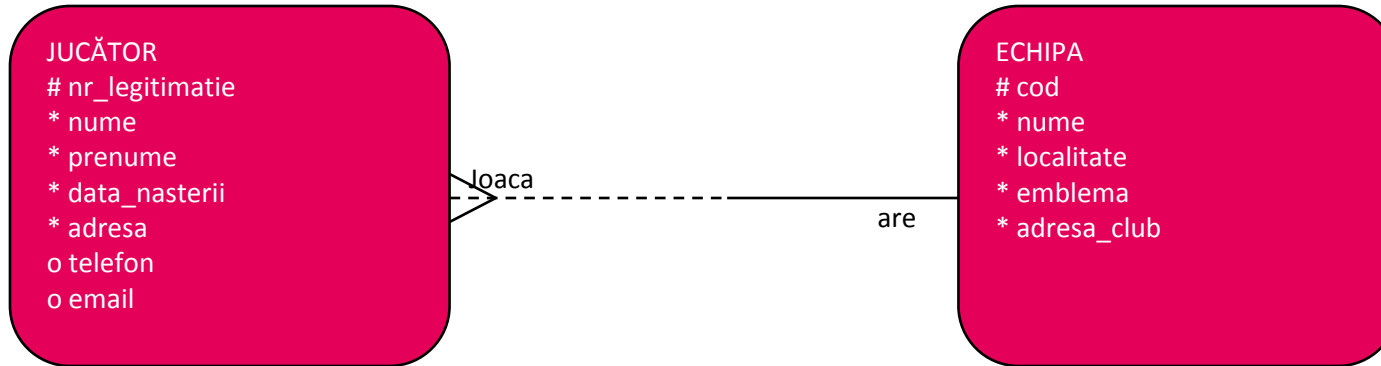
Regulile de integritate **garantează că datele introduse în baza de date sunt corecte și valide**. Aceasta înseamnă că dacă există orice regulă sau **restricție** asupra unei entități, atunci toate datele introduse în baza de date respectivă respectă aceste restricții. În Oracle, regulile de integritate se definesc la crearea tabelului folosind **constrângerile**.

Tipurile de reguli de integritate:

- **integritatea entităților** – indică faptul că nici o coloană ce face parte din cheia primară nu poate avea valoarea NULL și pentru fiecare înregistrare cheia primară trebuie să fie unică
- **integritatea de domeniu** – indică faptul că într-o coloană se introduc doar valori dintr-un domeniu anumit. De exemplu, putem impune ca nota unui elev să nu fie mai mică decât 1 și mai mare decât 10
- **integritatea referențială** – este o protecție care asigură ca fiecare valoare a cheii străine să corespundă unei valori a cheii primare din tabela referință. Integritatea referențială se poate încălca:
 - la adăugarea unei înregistrări în baza de date se poate încerca introducerea unor valori invalide pentru câmpurile cheii străine
 - la actualizarea bazei de date
 - la ștergerea unei înregistrări.



- Exemplu de încălcare a integrității referențiale: Entitățile JUCĂTOR și ECHIPA



- **cod** este cheie primară în ECHIPA și devine cheie străină în JUCĂTOR. Valoarea câmpului **cod** din cadrul tabelii JUCĂTOR corespunzătoare unui anumit jucător, trebuie să se găsească printre valorile câmpului **cod** din tabela ECHIPE, altfel, ar însemna că jucătorul joacă la o echipă inexistentă



4.10. PROGRAME DE VALIDARE ȘI DE ACȚIUNE

- La realizarea modelului conceptual al unei baze de date se ține cont de **modul în care funcționează afacerea** modelată, **datele** care trebuie să fie memorate, **relațiile** dintre acestea. Modul de utilizare a diferitelor date, modul în care acestea sunt relaționate, pot diferi de la o afacere la alta.
- **Regulile afacerii** unei organizații se referă în esență la procesele și fluxurile tuturor datelor și activităților zilnice din cadrul organizației.
- Regulile afacerii acoperă următoarele aspecte ale unei organizații:
 - orice tip de politici organizaționale de orice tip și la orice nivel al organizației
 - orice tip de formule de calcul (modul de calcul al salariilor, al ratelor la împrumuturi)
 - orice tip de reguli impuse de lege sau reguli interne ale organizației
- Regulile simple ale afacerii pot fi implementate în modelul bazei de date prin intermediul **relațiilor dintre entități**. Acest tip de reguli se numesc **reguli structurale**

- Alte reguli ale afacerii pot fi implementate folosind **regulile de integritate**. Există reguli pentru a căror implementare trebuie să scriem programe speciale folosind limbaje de programare. Aceste reguli se numesc **reguli procedurale**. În SQL acest tip de programe se numesc **declanșatoare** (triggere). Există două tipuri de declanșatoare:
 - **de aplicație** – care se execută când apar anumite evenimente
 - **ale bazei de date** care sunt lansate în executare când apar anumite evenimente asupra datelor (executarea unor comenzi ca INSERT, UPDATE, DELETE) sau apariția unor evenimente în sistem (logarea sau delogarea la/de la baza de date)
- Orice declanșator poate avea **rol de validare** a unei operații, **poate realiza diferite operații** suplimentare ca de exemplu, diferite calcule, caz în care se spune ca e un **declanșator de acțiune**.

În această lecție am învățat despre:

- Definirea modelelor de baze de date
- Caracteristicile principale ale bazelor de date relaționale
- Transformarea modelului conceptual al bazei de date (ERD) în modelul fizic (tabelele bazei de date)
- Principalele operații care se pot efectua asupra bazelor de date
- Rolul regulilor de integritate și care sunt acestea
- Programe de validare și acțiuni

SFÂRȘIT

