

ALEGEREA CHEILOR PRIMARE ÎNTR-O BAZĂ DE DATE – PROBLEME SPECIFICE ȘI CONSECINȚE

*Florin MIHAI, Ofelia ALECA,
Andrei STANCIU și Iuliana IONESCU*
Academia de Studii Economice din București, România

SELECTION OF PRIMARY KEYS IN A DATABASE
- SPECIFIC ISSUES AND CONSEQUENCES

ABSTRACT

This article discusses many general and particular aspects concerning the primary key selection in relational databases. Considering the essential role of the primary key in designing relational models, the article sustains that making the correct selection will influence not only the consistency and coherence of information, but also all other related concepts: foreign keys, normal forms and referential integrity. Therefore, choosing the best solution for primary keys will affect on long term the entire database model.

The paper identifies specific issues and makes comparative analysis for the main practices in primary key selection. Also, we expose the consequences that arise from choosing a primary key as it will be reflected in the future evolution of the database model.

🔑 *Databases, primary key, candidate key, surrogate key, foreign key, Normal Forms, Functional Dependency (FD)*

CONCEPTE

Așa cum este cunoscut, bazele *modelului relațional* au fost conturate de către matematicianul firmei IBM, E. F. Codd, în două articole de referință publicate în 1969 și 1970. Modelul relațional propune o organizare tabelară datelor stocate într-o bază de date, independentă de arhitectura hardware și software a sistemelor

și un set de operatori pentru extragerea datelor (Codd, 1969; 1970). Așa cum remarcă și Fotache (2005), deși teoria relațională propune termenul de “relație”, în practică s-a consacrat termenul de “tabelă”. Baza de date relațională este formată dintr-un ansamblu de tabele (relații) aflate în legătură.

Conform teoriei relaționale, o tabelă (relație) nu poate să conțină două sau mai multe rânduri (tupluri) identice. Fiecare rând al unei tabele trebuie să poată fi identificat într-o manieră clară, prin intermediul unui singur atribut sau a unui grup de atribute (composite key), ce aparțin tablei. În primul articol, Codd (1969), folosea termenul generic “cheie” pentru a sugera acest lucru. Abia în cel de-al doilea articol publicat de acesta (Codd, 1970), s-a folosit pentru prima dată termenul “*cheie primară*”, termen care astăzi ne este foarte familiar. Pe lângă caracteristica de unicitate amintită, o cheie primară mai trebuie să respecte două restricții:

- În cazul cheilor compuse, formate din mai multe atribute, nu se poate elimina un atribut parte din cheie, fără a distruge caracteristica de unicitate a tuplurilor unei tabele;
- Cheia primară nu admite sub nici o formă valori nule, iar în cazul cheilor compuse, nici un atribut parte din cheie nu poate avea valori nule. Conform uneia din regulile pe care trebuie să le respecte bazele de date relaționale, formulate de Codd (1985), valoarea nul este diferită de valoarea “zero” din punct de vedere numeric și de asemenea diferită de un șir de caractere de lungime zero. Practic, valoarea nul este independentă de tipul de date și se folosește atunci când informațiile care ar trebui stocate lipsesc.

O altă definiție asemănătoare cu prima, dar într-o manieră mai explicită, este dată de Chapple (2006). Cheia primară a unui tabel relațional identifică în mod unic fiecare înregistrare a unui tabel. Aceasta, poate fi formată dintr-un atribut al aceluși tabel ce are proprietatea de unicitate sau poate fi generată (Stanciu *et al.*, 2008) de către sistemul de gestiune al bazelor de date (de exemplu *Autonumber* pentru Microsoft Access sau *Guid* în Microsoft SQL Server).

Cheia primară se consideră a fi o *cheie naturală*, dacă atributul sau grupul de atribute care formează cheia, face parte din mulțimea atributelor ce caracterizează entitatea identificată de cheia primară. Valorile cheii naturale exprimă legăturile din lumea reală care există între aceasta și entitatea identificată. Dacă cheia primară este reprezentată de un atribut cu valori arbitrare, unice, dar care nu exprimă nicio legătură din lumea reală cu entitatea identificată, se consideră o *cheie surogat*. Sunt multe păreri pro și contra referitoare la alegerea unei chei primare naturale sau surogat. Având în vedere și aspectele prezentate de Ambler (2006) referitor la cheile naturale versus cheile surogat, în partea de final a articolului ne propunem să surprindem într-un studiu comparativ, avantajele și dezavantajele alegerii unei chei

primare naturale sau a unei chei primare surogat, ca parte importantă din practicile de alegere a unei chei primare.

Dependență funcțională. Fiind date două atribute D și E și o funcție $f: D \rightarrow E$. Spunem că f este o dependență funcțională simplă, dacă pentru $(\forall) d \in D$ distinct îi corespunde o singură valoare $f(d)=e \in E$.

O dependență funcțională $f: D \rightarrow E$ este elementară (completă), dacă nu există o altă dependență $g: D' \rightarrow E$, unde $D' \subset D$ adică nu mai poate fi descompusă astfel încât rezultatul obținut să fie tot o dependență funcțională.

O tabelă poate avea mai multe atribute sau mai multe grupuri de atribute, care să îndeplinească simultan condițiile pentru a fi cheie primară a tabelii. Acestea se numesc *chei candidate*, însă doar cea care este aleasă ca identificator al tabelii este denumită cheie primară. Cheile candidat se definesc conform (Date, 2004; Fotache, 2005; Florescu & Năstase, 2002) în felul următor:

Fie $R [A_1, A_2, \dots, A_n]$ o relație și K o submulțime a mulțimii de atribute ($K \subseteq \{A_1, A_2, \dots, A_n\}$). Atunci K este o cheie candidat pentru R dacă și numai dacă are următoarele caracteristici:

- **Unicitate:** nici o valoare valabila a variabilei R nu conține vreodată două tupluri distincte cu aceeași valoare a mulțimii K . $(\forall) A_i \notin K, A_i$ este dependent funcțional pe K . Adică există o dependență funcțională simplă $f: K \rightarrow R \setminus K$
- **Ireductibilitate:** nici o submulțime adecvata a mulțimii K nu are proprietatea de unicitate. $(\forall) K' \subset K, A_i$ nu este dependent funcțional pe K .

Pe baza conceptului de dependență funcțională se bazează și cele trei *forme normale*, formulate de Codd, pe care trebuie să le respecte tabelii unei baze de date relaționale:

- O relație este în *forma normală 1 (1NF)*, dacă și numai dacă toate attributele ei conțin numai valori atomice, care nu mai pot fi descompuse.
- O relație este în *forma normală 2 (2NF)* dacă și numai dacă este în 1NF și orice atribut noncheie se află în dependență funcțională completă față de cheia primară.
- O relație este în *forma normală 3 (3NF)* dacă și numai dacă este în 2NF și fiecare atribut noncheie nu se află în dependență funcțională tranzitivă față de cheia primară.

Termenul *foreign key*, folosit de Codd în articolul din 1970, se referă la attributele sau grupurile de atribute care pun în legătură rândurile unei tabelii cu rândurile altei tabelii. În cazuri excepționale, pot exista chei externe care pun în legătură rândurile unei tabelii cu ale ei însăși, privite dintr-o anumită perspectivă. Altfel spus *cheia externă* este un atribut sau un grup de atribute ale unui tabel definite sub formă de

cheie primară în alt tabel (sau chiar în același tabel) și servește pentru a defini legăturile dintre tabele.

Pe baza noțiunii de cheie externă, s-a dezvoltat conceptul de *restricție de integritate referențială*. O restricție de integritate referențială există atunci când, dacă valorile cheii externe nu au valori nule, acestea trebuie să fie neapărat dintre valorile cheii primare cu care se află în legătură. Conceptul de integritate referențială, este unul din conceptele fundamentale, care asigură coerența datelor stocate în baza de date.

1. ALEGEREA UNEI CHEII PRIMARE ALE CĂREI VALORI SUNT SUSCEPTIBILE DE MODIFICARE ÎN TIMP, INCLUSIV DE SCHIMBARE A ENTITĂȚII PE CARE O REFERĂ

De cele mai multe ori atunci când alegem cheia primară a unui tabel experiența își pune amprenta asupra deciziei. Deși la prima vedere pare ușor de stabilit care sunt cheile candidat, de multe ori alegerea uneia dintre acestea drept cheie primară poate fi un proces dificil. Dificultatea nu constă numai în decizia de a alege un atribut sau a unui grup de atribute în detrimentul altuia/altora ci și în a anticipa eventualele modificări ale înregistrărilor din acel tabel. De exemplu, să luăm cazul unei tabele ce ar conține înregistrări referitoare la autoturismele reparate în cadrul unui atelier de reparații. Într-o primă etapă se face o listă a atributelor necesare să fie reținute despre fiecare autoturism. În cazul nostru acestea ar putea fi: *număr înmatriculare, an fabricație, tip motorizare, serie motor, serie șasiu, culoare, greutate*, etc. La prima vedere, fără o analiză atentă am putea fi tentați să alegem drept cheie primară numărul de înmatriculare al autoturismului, având în vedere că la modul general, numerele de înmatriculare sunt valori unice și la un moment dat se referă la o singură mașină.

Figura 1. Tabela *Mașini* la momentul t_0

MASINI					
Nr. înmatriculare (PK)	Serie șasiu	Serie motor	An fabricație	Culoare	Greutate
B-11-POL	AA1234	MOT2222	1990	alb	1200
B-22-ABC	BB2345	MOP3333	2000	gri metalizat	1350
B-33-DDD	VV5555	MMI7777	2005	roșu	850

Dar în timp, este posibil ca autoturismul să își schimbe numărul de înmatriculare (de exemplu la revânzare) și astfel atributul număr de înmatriculare nu identifică în mod unic o înregistrare, nefiind astfel satisfăcute condițiile pentru ca acest atribut să fie cheie primară. Practic, celelalte atribute ale tabelii nu se află în dependență funcțională față de atributul *Nr. înmatriculare* adică tabela nu respectă nici măcar NF2 (Figura 2).

Figura 2. Tabela Mașini la momentul t_1

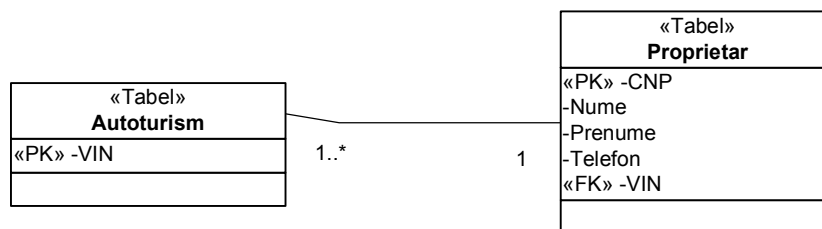
TABEL MASINI

Nr. înmatriculare (PK)	Serie șasiu	Serie motor	An fabricație	Culoare	Greutate
B-11-POL	AA1234	MOT2222	1990	alb	1200
B-22-ABC	BB2345	MOP3333	2000	gri metalizat	1350
B-33-DDD	VV5555	MMI7777	2005	roșu	850
B-33-DDD	KK9999	MMI7778	2008	alb	1250

Nici măcar seria șasiului sau seria motorului nu ar constitui o alegere corectă, deoarece există posibilitatea ca șasiul unei mașini să fie mutat pe o altă mașină (prin recarosare) sau în urma unui accident, motorul unei mașini să fie folosit pe o alta. În ambele cazuri, cele două serii ar identifica în timp două sau mai multe entități diferite. Dacă analizăm dependențele care există între atributele tabelului din Figura 2, constatăm că de fapt nu există nicio dependență funcțională în cadrul acesteia. Cu alte cuvinte nu are nicio cheie candidat, ceea ce înseamnă de fapt că nu are o cheie primară validă așa cum se prezintă în Figura 2.

Există însă conceptul de VIN (*Vehicle Identification Number* - numărul de identificare al vehiculului), acesta fiind un număr alocat de către producătorul auto, ca număr unic de identificare a unui autovehicul. Numerele sunt alocate pe baza standardului ISO 3779 - 1983, care permite identificarea universală, oriunde pe glob. VIN este format din caractere alfanumerice, cu ultimele patru fiind aproape întotdeauna numere, valabilă pe toată durata de viața a autoturismului, chiar dacă se schimbă proprietarul sau se înlocuiesc anumite componente.

Figura 3. Prezentarea modelului



Rămân problemele legate de istoricul numărului de înmatriculare și al celorlalte atribute ale căror valori se pot schimba în timp.

2. ALEGEREA UNEI CHEI PRIMARE CU VALORI CARE SE POT SCHIMBA ÎN TIMP, DAR CARE REFERĂ MEREU ACEEAȘI ENTITATE

Extinderea utilizării sistemelor informatice către toate domeniile, face ca multe din bazele de date din aplicațiile informatice să stocheze date despre persoane sub forma de angajați, clienți, furnizori, utilizatori ai sistemului, etc. Pentru a răspunde cerințelor de non redundanță și integritate a datelor, identificarea persoanelor în cadrul bazelor de date a fost și este amplu dezbătută. Dacă ne limităm doar la România, în cazul persoanelor juridice identificarea se face pe baza documentelor de înregistrare la Registrul Comerțului. Pentru persoanele fizice (ce nu au personalitate juridică) identificarea se poate face pe baza cărții de identitate (INEP, 2005) (document de identitate) sau a pașaportului.

Dacă luăm în considerare o tabelă simplificată *Persoane*, cu atributele *NrCarteIdentitate*, *SerieCarteIdentitate*, *CNP*, *Nume*, *Prenume*, ar putea exista două chei candidate: una compusă $\{NrCarteIdentitate, SerieCarteIdentitate\}$ și o alta formată dintr-un singur atribut: *CNP*.

Figura 4. Tabela *Persoane* cu cheia primară
 $\{NrCarteIdentitate, SerieCarteIdentitate\}$

PERSOANE

NrCarteIdentitate, SerieCarteIdentitate (PK)		CNP	Nume	Prenume
1234	AA	1720102510033	Ion	Marian
2345	BB	1750102510033	Mihai	Mircea
1234	BB	1760102510044	Tudor	Dan
2345	AA	1800122510033	Mihalache	Daniel

Cheia compusă $\{NrCarteIdentitate, SerieCarteIdentitate\}$ asigură identificarea în mod unic a rândurilor tabelii, însă valorile acesteia se pot schimba în decursul vieții persoanei. Ca să nu mai amintim că până la vârsta de 14 ani, în România o persoană nici nu are carte de identitate. Presupunând totuși că ne interesează doar persoanele de peste 14 ani, combinația $\{NrCarteIdentitate, SerieCarteIdentitate\}$ ridică doar probleme de păstrare a istoricului acestor valori în cazul în care se schimbă. Schimbarea valorilor acestor atribute poate surveni ca urmare a expirării la termenul legal a cărții de identitate, în urma pierderii, furtului etc.

Figura 5. Tabela *Persoane* cu cheia primară
 {NrCarteIdentitate, SerieCarteIdentitate}
 în cazul în care o persoană și-a schimbat cartea de identitate

NrCarteIdentitate, SerieCarteIdentitate (PK)		CNP	Nume	Prenume
1234	AA	1720102510033	Ion	Marian
2345	BB	1750102510033	Mihai	Mircea
1234	BB	1760102510044	Tudor	Dan
2345	AA	1800122510033	Mihalache	Daniel
4444	AA	1800122510033	Mihalache	Daniel

În acest caz dependențele funcționale următoare par a fi corecte pentru NF2:

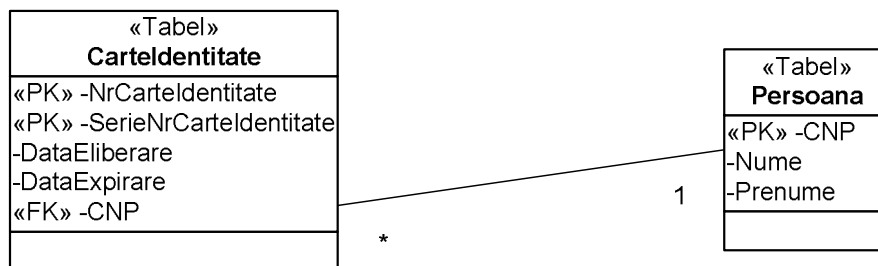
{NrCarteIdentitate, SerieNrCarteIdentitate} → CNP

{NrCarteIdentitate, SerieNrCarteIdentitate} → Nume

{NrCarteIdentitate, SerieNrCarteIdentitate} → Prenume

Rămân însă aspectele legate de istoricul acestor valori și suplimentar cele legate de perioada de valabilitate. O soluție mai bună decât cea prezentată în Figura 5 ar fi:

Figura 6. Modelul ce permite istoricul cărților de identitate reprezentat cu ajutorul diagramei claselor din UML



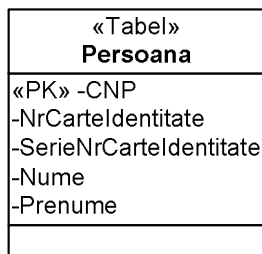
Păstrarea istoricului pentru valorile atributelor care se schimbă în timp ține mai mult de contextul în care se situează baza de date, dacă prezintă importanță memorarea istoricului acestor valori sau nu.

Codul Numeric Personal (CNP) constituie numărul de ordine atribuit de Evidența Populației unui individ la naștere. Codul numeric personal reprezintă un număr semnificativ ce individualizează o persoană fizică și constituie singurul

identificator pentru toate sistemele informatice care prelucrează date nominale privind persoanele fizice din România. Practic un cod numeric personal se atribuie la nașterea unei persoane și se păstrează pe tot parcursul vieții acesteia.

Soluția din Figura 4, Figura 5 și din Figura 6, poate fi exprimată mult mai comod așa cum se observă din Figura 7.

Figura 7. Reprezentarea tabelului *Persoana* în UML

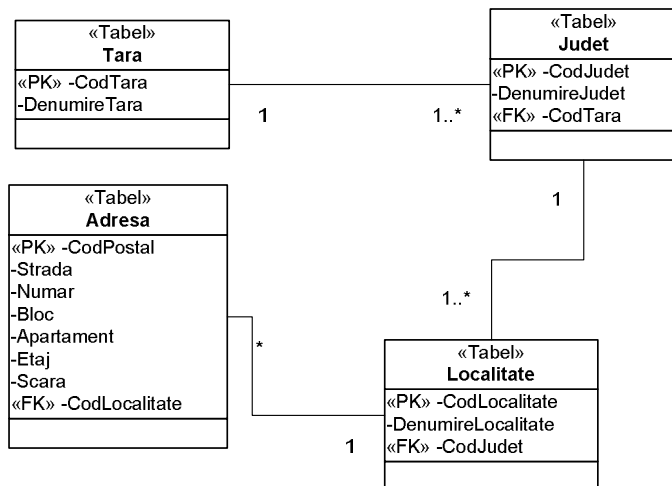


3. ALEGEREA UNEI CHEI PRIMARE CU VALORI CARE NU SE SCHIMBA ÎN TIMP

Identificarea adresei este un alt exemplu tipic destul de delicat atunci când se pune problema de alegere a cheii primare. Ținând cont de specificul fiecărei țări, dar în contextul globalizării, trebuie să găsim o modalitate de a identifica localitățile și în cadrul acestora adresele. Prima problemă este reprezentarea ierarhiei acestora începând cu cele mai mici unități administrative teritoriale cum ar fi satul sau orașul care fac parte dintr-un județ sau regiune care se situează într-o anumită țară, într-o anumită regiune a unui continent s.a. În cazul țărilor lucrurile sunt destul de clare, existând reglementări internaționale în vigoare (ISO, 1997) prin care fiecare țară este codificată printr-un cod de două caractere sau prin unul de trei caractere sau printr-un cod numeric. Tabelul ce conține județele sau statele este static adică modificările ce ar putea surveni în timp în cadrul tabelului sunt destul de reduse. O cheie primară naturală pentru această tabelă ar putea fi codificările internaționale standardizate de către ISO 3166-2 (în cazul României ISO 3166-2: RO). Aceste coduri sunt formate din cinci caractere: două reprezintă codul țării urmate de semnul minus și apoi de alte caractere specifice județului (de exemplu RO-AB, FR-25, US-AK, GB-ABD). Existența cheii externe cod Țară în cadrul tabelului Județ ar face ca introducerea valorilor codului de județ în formatul specificat de către ISO 3166-2 să fie redundant. Simplificarea codului prin păstrarea codurilor specifice regiunilor fiecărei țări nu garantează proprietatea de unicitate a cheii primare deoarece pot exista regiuni din țări diferite codificate în același fel. Deci, pentru alegerea cheii primare a tabelului Județ am avea două chei candidat, codul alocat de ISO 3166-2 specific fiecărei țări

și o cheie surogat (care este mereu o cheie candidat). Identificarea localităților în cadrul județelor poate să fie făcută cu ajutorul unei chei surogat.

Figura 8. Model de reprezentare a adreselor realizat cu ajutorul diagramei claselor din UML



4. ALEGEREA UNEI CHEI PRIMARE SUROGAT?

Alegerea unei chei surogat este influențată de mai mulți factori:

- **Existența atributelor autoincrementale.** Majoritatea celor mai importanți furnizori de baze de date - firme cum ar fi Oracle, Sybase, Microsoft - pun în aplicare o cheie surogat numită cheie incrementală. De fiecare dată când un rând este creat valoarea cheie pentru acel rând se incrementează. Valorile pot să fi unice în cadrul bazei de date sau în cadrul tabelului. Existența acestor attribute scutește utilizatorul de anumite artificii pentru atribuirea valorii unei chei candidat.
- **Modalitatea de întreținere.** Cheile surogat nu exprimă nicio legătură din lumea reală modelată, fiind din acest motiv mai ușor de întreținut.
- **Dinamica înregistrărilor în timp.** Din studii reiese că multe sisteme informatice devin ineficiente (unele chiar de la lansare în execuție) din cauza modificărilor (legislative și de altă natură) survenite în timp. O cauză a acestui eșec o poate constitui proiectarea deficitară a bazei de date și în special alegerea cheilor primare. Identificarea tabelelor importante (a celor ce au multe tabele copil sau a celor suspecte de schimbări în timp) cu chei primare surogat poate îmbunătăți șansele de succes ale aplicațiilor.

Cheile surogat sunt indicate a fi folosite și în cazul în care cheia primară este compusă din mai multe atribute. De exemplu dacă s-ar defini combinația de elemente {CodFz, OrasFurnizor} drept cheie candidat pentru o tabelă Furnizori, sistemul nu ar impune constrângerea ca numele furnizorilor sa fie unice „global” ci ar impune doar constrângerea ca numele sa fie unic local în cadrul localităților. Indicat ar fi să se atribuie furnizorilor coduri unice la nivel „global” deoarece existența unei chei primare compuse este mai greu de gestionat și de implementat în multe sisteme de gestiune a bazelor de date.

CONCLUZII

Deși conceptul de cheie primară este unul simplu, procesul de selectare a atributelor ce vor servi într-o primă fază drept chei candidat și apoi selectarea unei chei primare dintre acestea, poate fi un proces destul de dificil și greoi în multe cazuri. Alegerea acestora trebuie să țină cont de specificul activității, de anumite constrângeri și nu în ultimul rând de „flerul” și experiența echipei care realizează proiectarea unui sistem informatic. Factorul timp reprezintă de asemenea un element de bază care trebuie avut în vedere atunci când se proiectează o bază de date și de ce nu, aspecte legate de tendința de globalizare care se manifestă la nivel mondial. Probleme informatice legate de bazele de date care până recent puteau avea o rezolvare rapidă și rezonabilă, au devenit complicate odată cu fenomenul de globalizare și integrare a României în UE. Dacă luăm în considerare o tabelă cu studenți dintr-un sistem informatic al unei universități din România, constatăm rapid că nu putem stabili drept cheie primară a tabelii CNP-ul studentului și nici alegerea altor atribute/combinații de atribute nu ne garantează o soluție valabilă pe termen lung. De ce? Răspunsul este simplu. Odată cu aderarea la Uniunea Europeană se pot înscrie la universități din România persoane din tot spațiul Uniunii fără a necesita permise speciale de ședere în țara noastră, sau a altor documente care să presupună alocarea unor CNP-uri specifice străinilor. Sunt țări care nu dispun de CNP sau chiar dacă dispun, acestea sunt stabilite după reguli care nu îți garantează unicitatea CNP-urilor în contextul mai larg al studenților care provin nu doar dintr-o singură țară, ci din mai multe. Aspecte de acest gen ne conduc până la urmă în foarte multe cazuri către chei primare surogat pentru a ne pune la adăpost de surprizele oferite de factorul timp.

Soluția adoptării de chei primare surogat care nu au legătură cu realitatea, rămâne până la urmă o soluție comodă de a elimina potențiale probleme cu unele tabele pe termen lung.

BIBLIOGRAFIE

- Ambler, S. W. (2006) "Choosing a Primary Key: Natural or Surrogate?", disponibil on-line la <http://www.agiledata.org/essays/keys.html#Comparison>
- Chapple, M. (2006) „Primary Key Definition”, disponibil on-line la <http://databases.about.com/mbiopage.htm>
- Codd, E. F. (1969) "Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks ", *IBM Research Report RJ599*, IBM, San Jose, California, June
- Codd, E. F. (1970) "A Relational Model of Data for Large Shared Data Banks", *Communications of the ACM*, vol. 13, no. 6: 377-387
- Codd, E. F. (1985) "Is Your DBMS Really Relational?", *ComputerWorld*, October 14
- Codd, E. F. (1985) "Does Your DBMS Run By the Rules?", *ComputerWorld*, October 21
- Date, C.J. (2004) *An Introduction to database systems*, Peaterson Education, Inc.
- Florescu, V. & Năstase, P. (2002) *Baze de date : fundamente teoretice și practice*, București: Infomega
- Fotache, M. (2005) *Proiectarea bazelor de date*, Iași: Polirom
- INEP (2005) OUG nr.97/2005 disponibil la www.evidentapersoanelor.ro
- ISO (1997) Codes for the representation of names of countries and their subdivisions – Part 1: Country codes, ISO 3166-1
- Stanciu, A., Năstase, P. & Mihai, F. (2008) *Introducere in SQL Server 2008*, București: Infomega
- www.dreptonline.ro