Normalizarea datelor – Tutorial introductiv

🎯 Ce este normalizarea datelor

Normalizarea datelor este procesul prin care structura unei baze de date este organizată pentru a elimina redundanțele, a menține integritatea și a simplifica actualizarea datelor. Scopul este ca fiecare informație să fie stocată o singură dată, în locul potrivit.

🧱 Exemplu introductiv

Considerăm o bază de date pentru o școală. În tabelul de mai jos observăm repetiții inutile:

ElevID	NumeElev	Clasa	ProfesorDiriginte	Materie și Nota
1	Andrei Pop	10A	Dna X	TIC - 10
1	Andrei Pop	10A	Dna X	Matematică - 9
2	Maria Ionescu	10A	Dna X	TIC - 8

Se observă repetiția numelui profesorului. Dacă profesorul se schimbă, trebuie modificat în mai multe locuri, ceea ce poate duce la erori.

◇ Prima formă normală (1NF)

Regulă: fiecare celulă trebuie să conțină o singură valoare atomică, fără liste sau grupuri. Tabelul nostru respectă deja această regulă.

◇ A doua formă normală (2NF)

Regulă: toate atributele non-cheie trebuie să depindă de cheia completă. Pentru că identificatorul este compus (ElevID + Materie), iar unele câmpuri depind doar de ElevID, împărțim tabelul în două.

🧩 Tabela ELEV

ElevID	NumeElev	Clasa	ProfesorDiriginte
1	Andrei Pop	10A	Dna X
2	Maria Ionescu	10A	Dna X

🧩 Tabela NOTE

ElevID	Materie	Nota
1	TIC	10
1	Matematică	9
2	TIC	8

◇ A treia formă normală (3NF)

Regulă: elimină dependențele tranzitive. ProfesorDiriginte depinde de Clasa, nu de ElevID. Deci vom separa într-o tabelă CLASE.

🧩 Tabela CLASE

Clasa	ProfesorDiriginte
10A	Dna X

Avantajele normalizării

✅ Elimină redundanțele și economisește spațiu

✅ Evită erorile de actualizare

✅ Simplifică interogările și întreținerea bazei de date

Activitate practică

Creează o bază de date „Bibliotecă” cu câmpurile: CodCititor, NumeCititor, Carte, Autor, DataImprumutului. Normalizează baza de date până la forma 3NF, separând tabelele pentru CITITOR, CARTE și IMPRUMUT.

Concluzie

O bază de date bine normalizată este ca o bibliotecă bine organizată: fiecare informație are locul ei potrivit.

Normalizarea datelor – Alte explicații și exemple complete

Normalizarea datelor este procesul de organizare a datelor într-o bază de date pentru a reduce redundanța și a îmbunătăți integritatea acestora. Scopul este ca fiecare informație să fie stocată o singură dată, iar structura bazei să fie logică și eficientă.

◇ Exemplul de bază – Baza de date a unei biblioteci

Considerăm o bază de date simplă pentru o bibliotecă. Începem cu o singură tabelă care conține toate informațiile despre împrumuturi:

CodCititor	NumeCititor	Carte	Autor	DataÎmprumutului
C1	Ion Popescu	Enigma Otiliei	George Călinescu	10.10.2025
C1	Ion Popescu	Baltagul	Mihail Sadoveanu	11.10.2025
C2	Maria Ionescu	Enigma Otiliei	George Călinescu	09.10.2025

Se observă că numele cititorilor și al autorilor se repetă de mai multe ori. Aceasta duce la redundanță (date care se repetă) și la posibile inconsistențe.

◇ Prima formă normală (1NF)

O bază de date este în prima formă normală dacă toate câmpurile conţin doar valori atomice (adică o singură valoare per celulă). În exemplul nostru, fiecare celulă conţine o valoare unică, aşadar 1NF este îndeplinită.

◇ A doua formă normală (2NF)

Pentru a trece la a doua formă normală, eliminăm dependenţele parţiale. Observăm că NumeCititor depinde doar de CodCititor, nu de carte. Prin urmare, vom separa informaţiile despre cititori într-o tabelă distinctă.

🧩 Tabela CITITOR

CodCititor	NumeCititor
C1	Ion Popescu
C2	Maria Ionescu

🧩 Tabela IMPRUMUTURI

CodCititor	Carte	DataÎmprumutului
C1	Enigma Otiliei	10.10.2025
C1	Baltagul	11.10.2025
C2	Enigma Otiliei	09.10.2025

◇ A treia formă normală (3NF)

A treia formă normală elimină dependenţele tranzitive, adică situaţiile în care un atribut non-cheie depinde de alt atribut non-cheie. În cazul nostru, Autorul depinde de Carte, nu de CodCititor. Prin urmare, vom crea o tabelă separată pentru cărţi.

🧩 Tabela CARTE

CodCarte	Titlu	Autor
B1	Enigma Otiliei	George Călinescu
B2	Baltagul	Mihail Sadoveanu

🧩 Tabela IMPRUMUTURI (actualizată)

CodCititor	CodCarte	DataÎmprumutului
C1	B1	10.10.2025
C1	B2	11.10.2025
C2	B1	09.10.2025

Acum, toate informațiile sunt separate logic: CITITOR conține informațiile despre persoane, CARTE conține detaliile despre cărți, iar IMPRUMUTURI le leagă între ele prin coduri. Aceasta este o bază de date normalizată până la forma 3NF.

🧠 Alte exemple de normalizare

1️⃣ Exemplu – Magazin online:
- Tabela inițială conține: CodComanda, NumeClient, Produs, Preț, Adresă.
- Se observă că Adresa și NumeClient depind doar de client, nu de comandă. Se creează o tabelă CLIENT și o tabelă COMANDĂ.

2️⃣ Exemplu – Școală:
- Tabela inițială: Elev, Clasa, Diriginte, Materie, Nota.
- Dirigintele depinde de Clasa, iar Materia și Nota formează o relație separată. Rezultă tabelele ELEV, CLASA, NOTE, MATERIE.

3️⃣ Exemplu – Spital:
- Tabela inițială: Pacient, Doctor, SpecializareDoctor, Diagnostic, DatăConsultație.
- Specializarea depinde doar de Doctor, deci se separă în două tabele: DOCTOR și CONSULTAȚIE.

🌈 Avantajele normalizării

✅ Elimină redundanțele și economisește spațiu de stocare
✅ Evită anomaliile de actualizare (modificări incoerente)
✅ Permite integritatea și coerența datelor
✅ Ușurează interogările SQL și mentenanța bazei de date

💡 Concluzie

Normalizarea este esențială pentru o bază de date bine structurată. Prin aplicarea succesivă a celor trei forme normale, se obține o bază coerentă, fără repetiții, clară și ușor de extins. O bază de date normalizată este eficientă, logică și sigură.

Formele normale avansate – BCNF și 4NF

🎯 După aplicarea formelor normale de bază (1NF, 2NF, 3NF), există situații mai complexe în care pot apărea alte tipuri de dependențe. Pentru a elimina complet redundanțele, folosim formele normale avansate: Forma Normală Boyce–Codd (BCNF) și A Patra Formă Normală (4NF).

🔷 Forma Normală Boyce–Codd (BCNF)

📘 BCNF este o versiune mai strictă a celei de-a treia forme normale (3NF). O tabelă se află în BCNF dacă, pentru fiecare dependență funcțională $X \rightarrow Y$, X este o cheie candidat. Cu alte cuvinte, fiecare determinant trebuie să fie o cheie.

🧩 Exemplu – BCNF

Considerăm tabela PROGRAME_STUDENTI:

Student	Program	Profesor
Ana	Info	Popescu
Ion	Mate	Popescu
Maria	Info	Ionescu

Reguli:
- Un profesor predă un singur program (Profesor → Program)
- Un student este înscris la un singur program (Student → Program)

Cheia primară aparentă este (Student, Profesor), dar avem o dependență nepermisă: Profesor → Program. Pentru a aduce tabela în BCNF, o separăm în două:

1️⃣ Tabela PROFESOR

Profesor	Program
Popescu	Mate
Ionescu	Info

2️⃣ Tabela STUDENT

Student	Profesor
Ana	Popescu
Ion	Popescu
Maria	Ionescu

✅ Acum fiecare determinant este o cheie candidat, deci baza de date este în BCNF. Am eliminat toate dependențele funcționale nepermise.

🔷 A Patra Formă Normală (4NF)

📗 4NF elimină un alt tip de redundanță numit *dependență multivaluată*. Aceasta apare atunci când un atribut este legat independent de mai multe alte atribute.

🧩 Exemplu – 4NF

Considerăm tabela ACTIVITATI_STUDENT:

Student	Club	LimbaStraina
Ana	Robotică	Engleză
Ana	Robotică	Franceză
Ana	Muzică	Engleză

Fiecare student poate face parte din mai multe cluburi și poate vorbi mai multe limbi, dar cluburile și limbile nu depind una de alta. Avem două dependențe multivaluate independente:

- Student →→ Club
- Student →→ LimbaStraina

Pentru a ajunge la 4NF, împărțim tabela în două:

1️⃣ Tabela STUDENT_CLUB
Student	Club
Ana	Robotică
Ana	Muzică

2️⃣ Tabela STUDENT_LIMBA
Student	LimbaStraina
Ana	Engleză
Ana	Franceză

✅ Astfel, fiecare relație este independentă, iar baza de date este acum în 4NF.

📊 Compararea formelor normale

Formă	Elimină	Tip de dependență	Condiție
3NF	Dependențe tranzitive	Non-cheie → Non-cheie	Atributele non-cheie depind doar de chei
BCNF	Dependențe funcționale stricte	Non-cheie → Cheie	Fiecare determinant este cheie
4NF	Dependențe multivaluate	Seturi independente	Elimină legături între seturi independente

🧠 Exercițiu practic

Tabelă: CURSURI_STUDENTI

Student	Curs	Profesor	Activitate
Ana	TIC	Popescu	Laborator
Ana	TIC	Ionescu	Seminar
Ion	Matematică	Popescu	Laborator

Sarcină:
1️⃣ Identifică dependențele funcționale și multivaluate.
2️⃣ Normalizează tabela pas cu pas până la 4NF.
3️⃣ Desenează schema finală (tabele + chei).

💬 Concluzie

Formele BCNF și 4NF sunt utilizate pentru a obține baze de date robuste și fără redundanțe ascunse. Ele sunt folosite în proiecte complexe, unde consistența datelor este critică. Prin aplicarea lor, baza de date devine clară, coerentă și ușor de întreținut.