**Universitatea din București**

**Facultatea de Matematică și Informatică**

Inginerie Software

Aplicații profesionale ale bazelor de date

orientate obiect

Partea I

Student: Ceaușescu Ciprian – Mihai

grupa 406

1. Descrierea modelului

În acest proiect este propus un model pentru o bază de date în care se poate realiza gestionarea unei librarii sau a unei bublioteci pentru închiriere de jocuri – video sau boardgames.

În model, entitățile au fost organizate în patru categorii, după cum urmează:

1. Joc
2. client
3. magazin
4. angajat.

**Joc** - grupul principal, conține diferite informații legate de jocurile disponibile pentru închiriere, precum tipul, categoria, producătorul, numărul de jucători recomandați, precum și numărul de jocuri disponibile.

**Client** - reprezintă grupul în care se regăsesc informații legate de acesta, dar și produsele pe care el le-a închiriat.

**Magazin** - conține informații despre evenimentele ce se pot desfășura în aceste magazine, precum și departamentele din acestea.

**Angajat** - sunt disponibile informații despre toți angajatii companiei, poziția pe care aceștia o ocupă, precum și contractul sau nivelul salarial.

Această librărie, asigură clienților săi o gamă variată de jocuri, pe care aceștia le pot închiria pentru 30 de zile, din oricare magazin dintre cele disponibile. Clienții pot returna jocul în oricare dintre locațiile disponibile ale bibliotecii, și pot participa la diferitele evenimente oranizate în cadrul acestora.

Fiecare client are asociat un cont în baza de date, fiind neapărat necesar să transmită datele sale de contact, pentru eventualele întârzieri ale returnării. Acestia pot, de asemenea, să dea o nota între 1 și 5 tuturor jocurilor pe care le-au inchiriat.

Jocurile disponibile în baza de date sunt stocate împreuna cu informații referitoare la

tipurile, categoriile, producătorii, categoriile de vârstă recomandate pentru acestea, dar și notele oferite de clienți. Jocurile fac parte dintr-un singur tip, categorie, sau producător, însă pot exista mai multe jocuri din fiecare dintre acestea. În magazine, pot fi organizate diferite evenimente pentru clienți, în care aceștia pot participa la competiții de jocuri de societate și, pot testa jocurile lor preferate. În magazine există diferite departamente care se ocupa de logistică, închiriere jocuri, sau de achiziția acestora. Toți angajații au asignat câte un contract unic, și au un pachet salarial care face parte dintr-o clasă salarială. Aceștia ocupă anumite poziții în departamentele din care fac parte.

1. Constrângerile modelului

Modelul prezentat în secțiunea anterioară respectă următoarele reguli:

* un joc poate fi de un singur tip.
* un joc poate avea un singur producător.
* un angajat are un singur contract de muncă.
* un angajat ocupă o singură poziție.
* într-un contract trebuie definit un singur salariu.
* data începerii contractului este mai mică decât data încheierii acestuia.
* valoarea salariului trebuie să fie un număr între 1000 și 9999, cu 2 zecimale.
* într-un magazin pot fi mai multe evenimente.
* unui client i s-a emis cel puțin o factură.
* data emiterii unei facturi este mai mică decât data limită.
* data de returnare a jocurilor este mai mică sau egală decât data limită.
* nota dintr-un review poate fi un număr între 1 și 5.

1. Entități: descriere, cheie primară

Entitățile acestui model de bază de date sunt următoarele:

* **JOC**: informații generale despre toate jocurile disponibile în magazine.

Cheia primară: joc\_id

Chei străine: joc\_tip\_id, joc\_categorie\_id, joc\_numar\_jucatori\_id, joc\_producator\_id.

* **JOC\_TIP**: în această tabelă sunt reținute informații despre tipurile de jocuri.

Cheia primară: joc\_tip\_id

Chei străine: nu are.

* **JOC\_CATEGORIE**: o listă a tuturor categoriilor din care poate face parte un joc.

Cheia primară: joc\_categorie\_id

Chei străine: nu are.

* **JOC\_NUMAR\_JUCATORI**: în această tabelă sunt informații legate de numărul de jucători

recomandați pentru fiecare joc.

Cheia primară: joc\_numar\_jucatori\_id

Chei străine: nu are.

* **JOC\_PRODUCATOR**: conține toți producătorii de jocuri.

Cheia primară: joc\_producator\_id

Chei străine: nu are.

* **JOC\_EVALUARE**: reprezintă notele, pe o scară de la 1 la 5, evaluate de către clienți pentru jocurile selecționate de către aceștia.

Cheia primară: joc\_evaluare\_id

Chei străine: client\_id, joc\_id.

* **CLIENT**: oferă informații generale despre fiecare client în parte, precum și adresa acestora de contact.

Cheia primară: client\_id

Chei străine: nu are.

* **FACTURA\_CLIENT**: în această tabelă sunt toate facturile emise unui client, precum și

datele în care jocurile au fost împrumutate sau returnate și care sunt trecute într-o factură.

Cheia primară: factura\_client\_id

Chei străine: client\_id.

* **IMPRUMUT\_JOC**: tabel asociativ în care sunt reținute jocurile incluse în facturi, care au fost împrumutate.

Cheia primară: imprumut\_joc\_id

Chei străine: joc\_id, joc\_magazin\_id și factura\_client\_id.

* **MAGAZIN**: informații legate despre magazine.

Cheia primară: magazin\_id

Chei străine: nu are.

* **JOC\_MAGAZIN**: conține informații legate de numărul de jocuri din fiecare magazin în parte.

Cheia primară: compusă, formată din joc\_id și magazin\_id.

Chei străine: nu are.

* **MAGAZIN\_EVENIMENT**: în această tabelă sunt informații despre evenimentele din magazine.

Cheia primară: magazin\_eveniment\_id

Chei străine: magazin\_id.

* **MAGAZIN\_DEPARTAMENT**: detalii despre departamentele din magazine.

Cheia primară: magazin\_departament\_id

Chei străine: magazin\_id.

* **ANGAJAT**: descrierea angajaților din magazine. Aceștia ocupă o poziție într-un departament, și le este asociat un contract.

Cheia primară: angajat\_id

Chei străine: magazin\_departament\_id, angajat\_pozitie\_id și angajat\_contract\_id.

* **ANGAJAT\_POZITIE**: în această tabelă sunt pozițiile ocupate în fiecare departament în parte.

Cheia primară: angajat\_pozitie\_id

Chei străine: nu are.

* **ANGAJAT\_CONTRACT**: tabelă cu toate contractele asociate angajaților ce conține datele de început și încheiate ale acestora.

Cheia primară: angajat\_contract\_id

Chei străine: angajat\_nivel\_salarial\_id.

* **ANGAJAT\_NIVEL\_SALARIAL**: informații legate de salariile angajaților.

Cheia primară: angajat\_nivel\_salarial\_id.

Chei străine: nu are.

1. Relații: descriere, cardinalitate

Relațiile dintre entități sunt următoarele:

* **joc**\_are\_**joc\_tip**: relație ce face legătura între joc și tipul acestuia.

Cardinalitatea minimă este 1:0, unui tip de joc nu îi este neapărat asociat un joc, dar un joc trebuie să aibă un tip de joc.

Cardinalitatea maximă este M:1, un joc trebuie să fie asociat unui singur tip, însă mai multe jocuri pot fi asociate aceluiași tip.

* **joc**\_apartine\_**joc\_categorie**: relație ce face legătura între joc și categoria acestuia.

Cardinalitatea minimă este 1:1, cel puțin o categorie este asociată unui joc, iar un joc

trebuie să aibă minim o categorie.

Cardinalitatea maximă este 1:M, un joc este asociat unei categorii categorii, iar mai multe jocuri pot fi asociate acelorasi categorii.

* **joc**\_creat\_**joc\_producator**: relație ce face legătura între joc și producătorul acestuia. Cardinalitatea minimă este 1:1, un joc are un singur producător, iar un producător este asociat unui singur joc.

Cardinalitatea maximă este 1:M, un joc are cel mult un producător, iar un producător este asociat mai multor jocuri.

* **joc**\_are\_**joc\_evaluare**: relație care leagă jocurile de evaluările lor.

Cardinalitatea minima este 0:0, un joc nu trebuie să aibă neapărat o evaluare, iar o evaluare nu trebuie sa fie dată neaparat unui joc.

Cardinalitatea maximă este 1:M, un joc poate avea cel puțin o evaluare, iar aceeași evaluare poate fi dată mai multor jocuri de către persoane diferite.

* **joc**\_are\_**joc\_numar\_jucatori**: relație care face legătura între joc și numărul de jucători

recomandat.

Cardinalitatea minimă este 1:0, un joc are un număr de jucători recomandat, însă un număr de jucători nu este neapărat asociat tuturor jocurilor.

Cardinalitatea maximă este 1:M, un joc are un număr de jucători, iar un număr de jucători este recomandat pentru mai multe jocuri.

* **client**\_scrie\_**joc\_evaluare**: relație ce face legătura între clienți și evaluările pe care

aceștia le-au dat jocurilor.

Cardinalitatea minimă este 0:0, un client nu trebuie neapărat să scrie o evaluare, iar o evaluare nu trebuie să fi fost scrisă neapărat de un client.

Cardinalitatea maximă este M:N, un client poate scrie mai multe evaluări, și aceeași evaluare poate fi scrisă de mai multi clienți. (Evaluare = notă pe o scară de la 1 la 5).

* **client**\_face\_**imprumut\_joc**\_cu\_**factura\_client**: relație care leagă clienții de facturile cu

jocurile pe care acestia le-au împrumutat, prin intermediul tabelului asociativ imprumut\_joc.

Cardinalitatea minimă este 1:1, un client trebuie să aibă cel puțin o factură cu cel puțin

un produs.

Cardinalitatea maxiam este 1:M, un client poate avea mai multe facturi cu mai multe produse, însă mai multi clienți nu pot avea aceeași factură.

* **joc**\_disponibil\_**magazin**: relație ce leagă jocurile de stocul din magazine.

Cardinalitatea minimă este 1:1, trebuie să fie cel puțin un joc ca să poată exista în baza de date, iar într-un stoc este cel puțin un joc.

Cardinalitatea maximă este 1:M, un joc poate face parte din mai multe stocuri din diferite magazine.

* **magazin**\_are\_**joc**: relație ce leagă magazinele de stocuri.

Cardinalitatea minimă este 1:1, într-un magazin trebuie să fie cel puțin un joc ce face parte dintr-un stoc, iar un stoc este cel puțin într-un magazin.

Cardinalitatea maxima este 1:M, într-un magazin pot fi mai multe jocuri ce fac parte din diferite stocuri, însă nu poate fi același stoc în mai multe magazine.

* **magazin**\_organizează\_**magazin\_eveniment**: relație ce face legătura între magazine și evenimentele din acestea.

Cardinalitatea minimă este 0:1, un magazin nu trebuie să aibă neapărat un eveniment asociat, însă un eveniment trebuie să fie organizat pentru un magazin.

Cardinalitatea maximă este M:N, un eveniment poate fi asociat mai multor magazine, iar mai multe evenimente pot fi asociate aceluiași magazin.

* **magazin**\_are\_**magazin\_departament**: relație ce leagă magazinele de departemente. Cardinalitatea minimă este 1:0, un departament nu se află neapărat într-un magazin, însă un magazin trebuie să aibă cel puțin un departament.

Cardinalitatea maximă este M:N, un magazin poate avea mai multe departamente, și un departament poate fi în mai multe magazine.

* **angajat**\_lucrează\_**magazin\_departament**: relație ce face legătura între departament și angajați.

Cardinalitatea minimă este 1:1, un angajat face parte dintr-un singur departament și într-un departament este cel puțin un angajat.

Cardinalitate maxima este 1:M, într-un departament pot fi mai mulți angajați, dar un angajat face parte dintr-un singur departament.

* **angajat**\_are\_**angajat\_pozitie**: relație ce face legătura între angajat și funcția acestuia. Cardinalitatea minima este 1:0, o funcție nu este neapărat deținută de un angajat, însă un angajat trebuie să aibă o funcție.

Cardinalitatea maximă este M:1, un angajat trebuie să dețină o singură funcție, iar o funcție poate fi deținută de mai mulți angajați.

* **angajat**\_are\_**angajat\_contract**: relație care leagă angajații de contractele individuale de muncă.

Cardinalitatea minimă este 1:1, un contract este asociat unui singur angajat, iar un angajat poate avea un singur contract.

Cardinalitatea maximă este 1:1, un contract este asociat cel mult unui angajat, iar un angajat poate avea cel mult un contract.

* **angajat\_contract**\_are\_**nivel\_salarial**: relație care leagă contractele de salariile angajaților.

Cardinalitatea minimă este 1:1, un contract trebuie să aibă un salariu, iar un salariu este asociat unui singur contract.

Cardinalitatea maximă este 1:1, un contract trebuie să aibă cel mult salariu, iar un salariu este asociat cel mult unui contract.

1. Atribute: descriere, tip, constrângeri

Atributele tabelelor sunt următoarele:

**JOC**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(128) | numele jocului |
| imagine | VARCHAR(256) | link către imagine |
| descriere | VARCHAR(500) | descrierea jocului |
| an\_publicare | INT | anul în care a fost publicat |
| joc\_categorie\_id | INT | FK – tabelul joc\_categorie |
| joc\_producator\_id | INT | FK – tabelul joc\_producator |
| joc\_tip\_id | INT | FK – tabelul joc\_tip |
| joc\_numar\_jucatori\_id | INT | FK – tabelul joc\_numar\_jucatori |

**JOC\_TIP**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_tip\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(45) | numele tipului de joc |
| descriere | VARCHAR(45) | descrierea tipului din care face parte un joc |

**JOC\_CATEGORIE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_categorie\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(45) | numele categoriei jocului |
| descriere | VARCHAR(45) | descrierea categoriei jocului |

**JOC\_NUMAR\_JUCATORI**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_numar\_jucatori\_id | INT | PK – autoincrement de la 1 |
| valoare | VARCHAR2 (2) | numele categoriei jocului |
| descriere | VARCHAR(45) | descrierea categoriei jocului |

**JOC\_PRODUCATOR**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_producator\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(45) | numele producatorului jocului |
| tara | VARCHAR(45) | țara de origine |

**JOC\_EVALUARE**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_evaluare\_id | INT | PK – autoincrement de la 1 |
| nota | NUMBER(1) | nota (1-5) pentru un joc |
| client\_id | INT | FK – tabelul client |
| joc\_id | INT | FK – tabelul joc |

**CLIENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| client\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(45) | numele clientului |
| telefon | NUMBER(10) | numărul de telefon |
| email | VARCHAR(50) | adresa de e-mail |
| strada | VARCHAR(30) | adresă - stradă |
| numar | INT | adresă - numār |
| oras | VARCHAR(30) | adresā - oraș |

**FACTURA\_CLIENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| factura\_client\_id | INT | PK – autoincrement de la 1 |
| data\_inceput | DATE | data la care a fost emisā facture |
| data\_sfarsit | DATE | data limitā pentru returnarea produselor |
| data\_returnare | DATE | data la care au fost returnate produsele |
| client\_id | INT | FK – tabelul client |

**IMPRUMUT\_JOC**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| imprumut\_joc\_id | INT | PK – autoincrement de la 1 |
| joc\_id | INT | FK – tabelul joc\_magazin |
| magazin\_id | INT | FK – tabelul joc\_magazin |
| factura\_client\_id | INT | FK – tabelul factura\_client |

**MAGAZIN**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| magazin\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(64) | numele magazinului |
| strada | VARCHAR(30) | adresă - stradă |
| numar | INT | adresă - numār |
| oras | VARCHAR(30) | adresā - oraș |

**JOC\_MAGAZIN**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| joc\_id | INT | PK – PK din tabelul joc |
| magazin\_id | INT | PK – PK din tabelul magazin |
| cantitate | INT | numărul de jocuri din magazin |
| disponibil | INT | 1 (implicit) = disponibil, 0 = indisponibil |

**MAGAZIN\_EVENIMENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| magazin\_eveniment\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(128) | numele evenimentului |
| descriere | VARCHAR(500) | descrierea evenimentului |
| data\_eveniment | DATE | data desfasurarea evenimentului |
| magazin\_id | INT | FK – tabelul magazin |
| joc\_id | INT | FK – tabelul joc |

**MAGAZIN\_DEPARTAMENT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| magazin\_departament\_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(128) | numele departamentului |
| magazin\_id | INT | FK – tabelul magazin |

**ANGAJAT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| angajat \_id | INT | PK – autoincrement de la 1 |
| nume | VARCHAR(128) | numele angajatului |
| cnp | NUMBER(13) | cnp angajat |
| strada | VARCHAR(30) | adresă - stradă |
| numar | INT | adresă - numār |
| oras | VARCHAR(30) | adresā - oraș |
| telefon | NUMBER(10) | numărul de telefon |
| magazin\_departament\_id | INT | FK – tabelul magazin\_departament |
| angajat\_contract\_id | INT | FK – tabelul angajat\_contract |
| angajat\_pozitie\_id | INT | FK – tabelul angajat\_pozitie |

**ANGAJAT\_POZITIE**

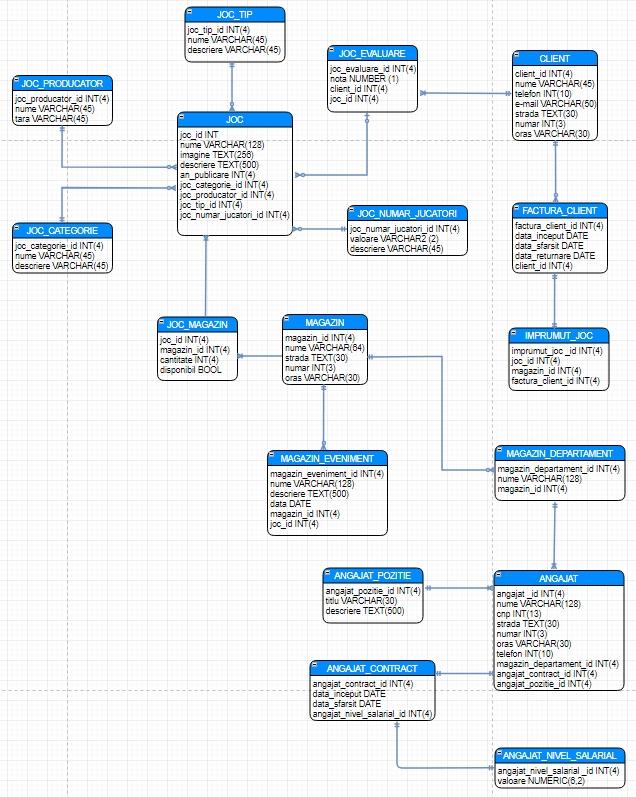
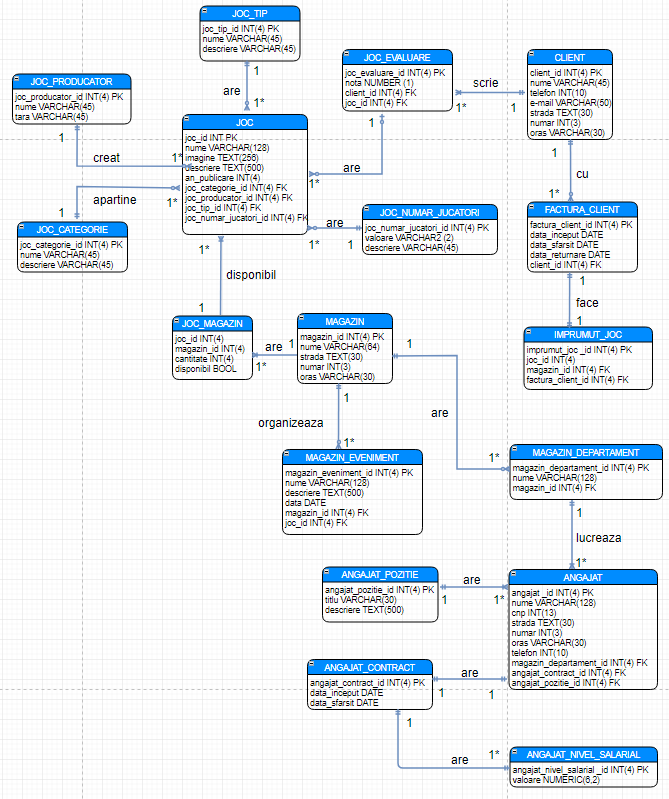
|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| angajat\_pozitie\_id | INT | PK – autoincrement de la 1 |
| titlu | VARCHAR(30) | numele poziției unui angajat |
| descriere | VARCHAR(500) | descrierea poziției unui angajat |

**ANGAJAT\_CONTRACT**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| angajat\_contract\_id | INT | PK – autoincrement de la 1 |
| data\_inceput | DATE | data la care a început contractul |
| data\_sfarsit | DATE | data la care se încheie contractul |
| angajat\_nivel\_salarial\_id | INT | FK – tabelul angajat\_nivel\_salarial |

**ANGAJAT\_NIVEL\_SALARIAL**

|  |  |  |
| --- | --- | --- |
| **Nume** | **Tip** | **Descriere** |
| angajat\_nivel\_salarial \_id | INT | PK – autoincrement de la 1 |
| valoare | NUMERIC(6,2) | suma exprimată în RON |

1. Diagrama entitate-relație
2. Diagrama conceptuală
3. Scheme relaționale

Schemele relaționale sunt următoarele:

* **joc**: #joc\_id, nume, imagine, descriere, an\_publicare, joc\_categorie\_id, joc\_producator\_id, joc\_tip\_id, joc\_numar\_jucatori\_id.
* **joc\_tip:** #joc\_tip\_joc, nume, descriere.
* **joc\_categorie:** #joc\_categorie\_id, nume, descriere.
* **joc\_numar\_jucatori:** #joc\_numar\_jucatori\_id, valoare, descriere.
* **joc\_producator:** #joc\_producator\_id, nume, tara.
* **joc\_evaluare:** #joc\_evaluare\_id, nota, client\_id, joc\_id.
* **client:** #client\_id, nume, telefon, email, strada, numar, oras**.**
* **factura\_client:** #factura\_client, data\_inceput, data\_sfarsit, data\_returnare, client\_id.
* **imprumut\_joc:** #imprumut\_joc\_id, joc\_id, magazin\_id, factura\_client\_id.
* **magazin:** #magazin\_id, nume, strada, numar, oras**.**
* **joc\_magazin:** #joc\_id, #magazin\_id, cantitate, disponibil.
* **magazin\_eveniment:** #magazin\_eveniment\_id, nume, descriere, data\_eveniment, magazin\_id, joc\_id.
* **magazin\_departament:** #magazin\_departament\_id, nume, descriere.
* **angajat:** #angajat\_id, nume, cnp, strada, numar, oras, telefon,

magazin\_departament\_id, angajat\_contract\_id, angajat\_pozitie\_id.

* **angajat\_pozitie:** #angajat\_pozitie\_id, titlu, descriere.
* **angajat\_contract:** #angajat\_contract\_id, data\_inceput, data\_sfarsit, angajat\_nivel\_salarial\_id.
* **angajat\_nivel\_salarial:** #angajat\_nivel\_salarial\_id, valoare.

1. Normalizare

Această bază de date a fost proiectată cu scopul de a reprezenta cât mai corect informațiile, iar posibilitatea de a ajunge la informații eronate să fie diminuată.

Pentru a ajunge la această performanță, au fost folosite diferite tehnici de normalizare. Aceasta presupune urmarea unor reguli prin care baza de date se poate normaliza până la un anumit grad. Dacă o cerință nu este satisfăcută, relația trebuie descompusă în mai multe relații, care individual satisfac anumite reguli de normalizare.

**Forma nenormalizată (UNF):** O tabelă care conține unul sau mai multe grupuri repetitive.

Tabelul **magazin\_departament** ar fi putut conține grupuri repetitive formate din numele angajaților dintr-un anumit departament, conform tabelului următor:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mag\_dep\_id | Nume | Descriere | Angajat | Angajat\_pozitie |
| 1 | HR | HR | Mihai, Ionel, Maria | Partener HR |
| 2 | IT | IT | Violeta, Vasile | Programatori |
| 3 | Contabilitate | Conta | George, Cristian | Contabili |

**Forma normală 1 (FN 1):** Exclude posibilitatea existenței grupurilor repetitive, cerând ca fiecare câmp într-o bază de date să cuprindă numai o valoare atomică. De asemenea, prima formă normală cere ca fiecare înregistrare să fie definită astfel încât să fie identificată în mod unic prin intermediul unei chei primare.

Tabelul de mai sus devine astfel:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mag\_dep\_id | Nume | Descriere | Angajat | Angajat\_pozitie |
| 1 | HR | HR | Mihai | Partener HR |
| 2 | HR | HR | Ionel | Partener HR |
| 3 | HR | HR | Maria | Partener HR |
| 4 | IT | IT | Violeta | Programatori |
| 5 | IT | IT | Vasile | Programatori |
| 6 | Contabilitate | Conta | George | Contabili |
| 7 | Contabilitate | Conta | Cristian | Contabili |

Similar în toate tabele create, câmpul **adresă**, care era un câmp cu valori multiple a fost împărțit în mai multe câmpuri, și anume: **stradă, număr, oraș**.

**Forma normal 2 (FN 2):** În FN 1 + A doua formă normală cere ca toate elementele unei tabele să fie dependente funcțional de totalitatea cheii primare. Dacă unul sau mai multe elemente sunt dependente funcțional numai de o parte a cheii primare, atunci ele trebuie să fie separate în tabele diferite. Dacă tabela are o cheie primară formată doar dintr-un atribut, atunci ea este automat în FN2 (a 2-a forma normală).

Tabelul **magazin\_departament** este împărțit acum în două tabele: **angajat** și **magazin\_departament**, după cum urmează:

**magazin\_departament:**

|  |  |  |
| --- | --- | --- |
| Mag\_dep\_id | Nume | Descriere |
| 1 | HR | HR |
| 2 | IT | IT |
| 3 | Contabilitate | Conta |

**angajat**:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ang\_id | nume | cnp | str | nr | ors | tel | mag\_dep\_id | ang\_contr\_id | an\_poz\_id |
| 1 | Cristi | 1.. | Lun | 1 | Cl | 004 | 1 | 1 | 3 |
| 2 | Mihai | 1.. | Ma | 2 | Sz | 004 | 2 | 2 | 7 |

În continuare, observăm faptul că multe tabele conțin câmpuri referitoare la **adresă,** și anume: **stradă, număr, oraș.** Pentru ca tabelele să fie în FN2, vom crea tabela adresă, care va stoca informații cu privire la adresele respectivelor entități.

Luăm un singur exemplu de acest tip, și împărțim tabela **angajat** în două tabele: **angajat** și **adresă**, urmând ca tabela **angajat** să aibe FK cheia tabelei **adresă**. Exemplul este similar pentru toate tabelele care au informații referitoare la adresă.

**angajat**:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ang\_id | nume | cnp | tel | mag\_dep\_id | ang\_contr\_id | ang\_poz\_id | adresa\_id |
| 1 | Cristi | 1.. | 004 | 1 | 1 | 3 | 1 |
| 2 | Mihai | 1.. | 004 | 2 | 2 | 7 | 2 |

**adresa**:

|  |  |  |  |
| --- | --- | --- | --- |
| adresa\_id | strada | numar | oras |
| 1 | Lunga | 1 | Calarasi |
| 2 | Magheru | 2 | Slobozia |

**Forma normal 3 (FN 3):** În FN 2 + Toate atributele non-chei ale unei relații depind numai de chei candidate ale acelei relații.

Un exemplu de transformări pe care le facem pentru a respecta FN 3 -

Tabelele de mai sus se impart în alte noi tabele, după cum urmează:

**magazin\_departament:**

|  |  |  |
| --- | --- | --- |
| Mag\_dep\_id | Nume | Descriere |
| 1 | HR | HR |
| 2 | IT | IT |
| 3 | Contabilitate | Conta |

**angajat**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ang\_id | nume | cnp | tel | mag\_dep\_id | ang\_poz\_id | adresa\_id |
| 1 | Cristi | 1.. | 004 | 1 | 3 | 1 |
| 2 | Mihai | 1.. | 004 | 2 | 7 | 2 |

**adresa**:

|  |  |  |  |
| --- | --- | --- | --- |
| adresa\_id | strada | numar | oras |
| 1 | Lunga | 1 | Calarasi |
| 2 | Magheru | 2 | Slobozia |

**angajat\_pozitie**:

|  |  |  |
| --- | --- | --- |
| angajat\_pozitie | titlu | Descriere |
| 3 | HR | HR |
| 7 | Programator | Programator web |

1. Crearea tabelelor în SQL

Începem prin a creat tabelele care nu au constrângeri de cheie străină:

1. CREATE TABLE adresa (

adresa\_id INT NOT NULL,

strada VARCHAR(45) NOT NULL,

numar VARCHAR(5) NOT NULL,

oras VARCHAR(45) NOT NULL,

PRIMARY KEY (adresa\_id));

1. CREATE TABLE joc\_tip (

joc\_tip\_id INT NOT NULL,

nume VARCHAR(45) NOT NULL,

descriere VARCHAR(45) NOT NULL,

PRIMARY KEY (joc\_tip\_id),

CONSTRAINT joc\_tip\_nume UNIQUE (nume));

1. CREATE TABLE joc\_categorie (

joc\_categorie\_id INT NOT NULL,

nume VARCHAR(45) NOT NULL,

descriere VARCHAR(45) NOT NULL,

PRIMARY KEY (joc\_categorie\_id),

CONSTRAINT joc\_categorie\_nume UNIQUE (nume));

1. CREATE TABLE joc\_numar\_jucatori (

joc\_numar\_jucatori\_id INT NOT NULL,

valoare VARCHAR(2) NOT NULL,

descriere VARCHAR(45) NOT NULL,

PRIMARY KEY (joc\_numar\_jucatori\_id),

CONSTRAINT joc\_numar\_jucatori\_valoare UNIQUE (valoare));

1. CREATE TABLE joc\_producator (

joc\_producator\_id INT NOT NULL,

nume VARCHAR(45) NOT NULL,

tara VARCHAR(45) NOT NULL,

PRIMARY KEY (joc\_producator\_id),

CONSTRAINT joc\_producator\_nume UNIQUE (nume));

1. CREATE TABLE client (

client\_id INT NOT NULL,

nume VARCHAR(45) NOT NULL,

telefon NUMBER(10) NOT NULL,

email VARCHAR(45) NOT NULL,

adresa INT NOT NULL,

PRIMARY KEY (client\_id),

FOREIGN KEY (adresa) REFERENCES adresa(adresa\_id));

1. CREATE TABLE factura\_client (

factura\_client\_id INT NOT NULL,

data\_inceput DATE NOT NULL,

data\_sfarsit DATE NOT NULL,

data\_returnare DATE NULL,

client\_id INT NOT NULL,

PRIMARY KEY (factura\_client\_id),

FOREIGN KEY (client\_id) REFERENCES client(client\_id));

1. CREATE TABLE joc\_evaluare (

joc\_evaluare\_id INT NOT NULL,

nota NUMBER(1) NOT NULL,

client\_id INT NOT NULL,

joc\_id INT NOT NULL,

PRIMARY KEY (joc\_evaluare\_id),

FOREIGN KEY (client\_id) REFERENCES client(client\_id),

FOREIGN KEY (joc\_id) REFERENCES joc(joc\_id));

1. CREATE TABLE joc (

joc\_id INT NOT NULL,

nume VARCHAR(128) NOT NULL,

imagine VARCHAR2(256) NOT NULL,

descriere VARCHAR2(500) NOT NULL,

an\_publicare INT NOT NULL,

joc\_categorie\_id INT NOT NULL,

joc\_producator\_id INT NOT NULL,

joc\_tip\_id INT NOT NULL,

joc\_numar\_jucatori\_id INT NOT NULL,

PRIMARY KEY (joc\_id),

FOREIGN KEY (joc\_categorie\_id) REFERENCES joc\_categorie(joc\_categorie\_id),

FOREIGN KEY (joc\_producator\_id) REFERENCES joc\_producator(joc\_producator\_id),

FOREIGN KEY (joc\_tip\_id) REFERENCES joc\_tip(joc\_tip\_id),

FOREIGN KEY (joc\_numar\_jucatori\_id) REFERENCES joc\_numar\_jucatori(joc\_numar\_jucatori\_id));

1. CREATE TABLE magazin (

magazin\_id INT NOT NULL,

nume VARCHAR2(64) NOT NULL,

adresa\_id INT NOT NULL,

PRIMARY KEY (magazin\_id),

FOREIGN KEY (adresa\_id) REFERENCES adresa(adresa\_id));

1. CREATE TABLE imprumut\_joc (

imprumut\_joc\_id INT NOT NULL,

joc\_id INT NOT NULL,

magazin\_id INT NOT NULL,

factura\_client\_id INT NOT NULL,

PRIMARY KEY (imprumut\_joc\_id),

FOREIGN KEY (joc\_id) REFERENCES joc(joc\_id),

FOREIGN KEY (magazin\_id) REFERENCES magazin(magazin\_id),

FOREIGN KEY (factura\_client\_id) REFERENCES factura\_client(factura\_client\_id));

1. CREATE TABLE joc\_magazin (

joc\_id INT NOT NULL,

magazin\_id INT NOT NULL,

cantitate INT NOT NULL,

disponibil INT NOT NULL,

PRIMARY KEY (joc\_id, magazin\_id),

FOREIGN KEY (joc\_id) REFERENCES joc(joc\_id),

FOREIGN KEY (magazin\_id) REFERENCES magazin(magazin\_id));

1. CREATE TABLE magazin\_eveniment (

magazin\_eveniment\_id INT NOT NULL,

nume VARCHAR(128) NOT NULL,

descriere VARCHAR(500) NOT NULL,

data\_eveniment DATE NOT NULL,

magazin\_id INT NOT NULL,

joc\_id INT NOT NULL,

PRIMARY KEY (magazin\_eveniment\_id),

FOREIGN KEY (joc\_id) REFERENCES joc(joc\_id),

FOREIGN KEY (magazin\_id) REFERENCES magazin(magazin\_id));

1. CREATE TABLE magazin\_departament (

magazin\_departament\_id INT NOT NULL,

nume VARCHAR(128) NOT NULL,

magazin\_id INT NOT NULL,

PRIMARY KEY (magazin\_departament\_id),

FOREIGN KEY (magazin\_id) REFERENCES magazin(magazin\_id));

1. CREATE TABLE angajat\_pozitie (

angajat\_pozitie\_id INT NOT NULL,

titlu VARCHAR(30) NOT NULL,

descriere VARCHAR(500) NOT NULL,

PRIMARY KEY (angajat\_pozitie\_id));

1. CREATE TABLE angajat\_nivel\_salarial (

angajat\_nivel\_salarial\_id INT NOT NULL,

valoare NUMERIC(6,2) NOT NULL,

PRIMARY KEY (angajat\_nivel\_salarial\_id));

1. CREATE TABLE angajat\_contract (

angajat\_contract\_id INT NOT NULL,

data\_inceput DATE NOT NULL,

data\_sfarsit DATE,

angajat\_nivel\_salarial\_id INT NOT NULL,

PRIMARY KEY (angajat\_contract\_id),

FOREIGN KEY (angajat\_nivel\_salarial\_id) REFERENCES angajat\_nivel\_salarial(angajat\_nivel\_salarial\_id));

1. CREATE TABLE angajat (

angajat\_id INT NOT NULL,

nume VARCHAR(128) NOT NULL,

cnp NUMBER(13) NOT NULL,

adresa INT NOT NULL,

telefon NUMBER(10) NOT NULL,

magazin\_departament\_id INT NOT NULL,

angajat\_contract\_id INT NOT NULL,

angajat\_pozitie\_id INT NOT NULL,

PRIMARY KEY (angajat\_id),

FOREIGN KEY (magazin\_departament\_id) REFERENCES magazin\_departament(magazin\_departament\_id),

FOREIGN KEY (angajat\_contract\_id) REFERENCES angajat\_contract(angajat\_contract\_id),

FOREIGN KEY (angajat\_pozitie\_id) REFERENCES angajat\_pozitie(angajat\_pozitie\_id),

FOREIGN KEY (adresa) REFERENCES adresa(adresa\_id);