

BAZE DE DATE: PROIECT BAZA DE DATE A UNEI BĂNCI

PROFESOR COORDONATOR:
VASILE SILVIU-LAURENȚIU

STUDENT:
FOTA ADRIAN-MIHAEL

Cuprins

INTRODUCERE.....	3
Motivarea alegerii temei de proiect	3
Software folosit.....	3
Prezentarea modelului din lumea reală	4
ENTITĂȚI, ATRIBUTE, RELAȚII, CONSTRÂNGERI.....	5
Diagrama Entitate-Relație (E-R)	5
Descrierea entităților și atributelor	6
Descrierea relațiilor dintre entități.....	8
Diagrama conceptuală	9
Prezentarea constrângerilor de integritate	10
Schemele relaționale	11
IMPLEMENTAREA ÎN MYSQL	12
Crearea tabelor și definirea constrângerilor	12
Inserarea datelor	17
INCHEIERE.....	19
Concluzii	19
Bibliografie	19

INTRODUCERE

Motivarea alegerii temei de proiect

Bazele de date au un rol esențial în gestionarea eficientă a cantităților mari de informații în mediul digital contemporan. În cadrul acestei evoluții tehnologice, am decis ca proiectul meu să se concentreze pe crearea și dezvoltarea unei simple baze de date bancare.

În calitate de clienți ai diverselor bănci, intrăm în contact frecvent cu infrastructura lor de baze de date și chiar facem parte din aceasta, adesea fără să conștientizăm acest aspect. Cu toții suntem familiarizați cu tranzacțiile bancare, verificarea soldurilor sau aplicațiile online ai băncilor, fără să ne gândim la mecanismele din spatele acestor lucruri. Astfel, am ales tematica bancară pentru proiectul meu deoarece este oarecum trivială, dar în același timp complexă și intricată.

Software folosit

Pentru implementarea bazei de date am ales să folosesc MySQL ca sistem de gestionare al bazelor de date (SGBD), codul fiind implementat în cadrul MySQL Workbench, un mediu de dezvoltare (IDE - Integrated Development Environment) dezvoltat de Oracle Corporation¹.

MySQL este un SGBD open-source, fiind considerat al 2-lea cel mai folosit SGBD după Oracle SQL², conform DB-Engines. MySQL este recunoscut pentru performanță și ușurință de integrare, fiind des folosit pentru dezvoltarea aplicațiilor web și în alte proiecte software. În plus, MySQL permite utilizarea enumerărilor (datatype enum) și oferă o modalitate simplă de interacțiune cu o interfață web prin pachetul “mysql” din Node Packet Manager (NPM).

MySQL Workbench este un mediu de dezvoltare specific MySQL care permite modelarea, dezvoltarea și gestionarea unei baze de date MySQL. Bazele de date și tabelele din cadrul acestora pot fi create și alterate atât folosind cod SQL cât și o interfață simplificată nativă aplicației. Pentru dezvoltarea proiectului meu, am folosit MySQL Workbench 8.0 CE (Community Edition).

¹ https://docs.oracle.com/cd/E17952_01/workbench-en

² <https://db-engines.com/en/ranking>

Prezentarea modelului din lumea reală

Modelul bazei de date este bazat pe unul existent și în lumea reală, deci este esențial să încep cu prezentarea modelului în contextul acesteia.

Proiectul reprezintă o bază de date a unei bănci ficționale, denumite “MyBank” și prezintă entitățile și relațiile corespunzătoare elementelor din lumea reală. O bancă are acces la datele clienților săi, precum CNP-ul, adresa de reședință și salariul. Clienții, la rândul lor, deschid conturi la bancă, iar fiecărui cont îi este adesea asociat unul sau mai multe carduri bancare. Cardurile bancare au active diferite oferte, iar clienții pot interacționa cu sistemul prin efectuarea tranzacțiilor sau accesarea aplicației online a băncii, toate acestea fiind salvate în baza de date bancară.

Modelul prezintă un set de reguli ce sunt reflectate și în proiect:

- Un client reprezintă o persoană fizică, de cel puțin 16 ani împliniți, ce folosește serviciile băncii.
- Clientul poate avea o singură cetățenie principală pe care o folosește în momentul primei interacțiuni cu banca.
- Clientul poate locui la o singură adresă, fiind luată în considerare adresa curentă de reședință. Nu există clienți care nu au o locuință.
- Banca reține o adresă simplificată a clientului (țara, orașul și strada).
- O persoană fizică poate fi clientul băncii fără a avea un cont deschis la banca respectivă.
- Un client poate avea mai multe conturi la aceeași bancă, pe care le poate folosi în scopuri diferite.
- Banca ficțională are sediul în România, și categorizează doar țările Uniunii Europene (orice țară în afara UE va fi pusă în categoria “not_eu”).
- Pentru a deveni clientul băncii nu este necesar ca persoana fizică să aibă un buletin românesc. În cazul cetățenilor străini se va folosi o alternativă la CNP (nr. de identificare specific țării respective).
- Ofertele active pentru cardurile bancare reprezintă o rată specifică de dobândă atunci când clientul are bani pe card pentru un an sau mai mult. Oferta activă pentru un card nu depinde de vârsta clientului, salariul acestuia sau tipul cardului.
- Banca este denumită “MyBank” și are identificatorul IBAN MBNK.
- Un client poate încerca să interacționeze cu aplicația online a băncii chiar dacă nu are un cont deschis. Deși clientul nu poate să interogheze soldul contului său deoarece acesta nu există, încercarea respectivă va fi detectată și salvată în istoricul activităților online.

ENTITĂȚI, ATRIBUTE, RELAȚII, CONSTRÂNGERI

Diagrama Entitate-Relație (E-R)

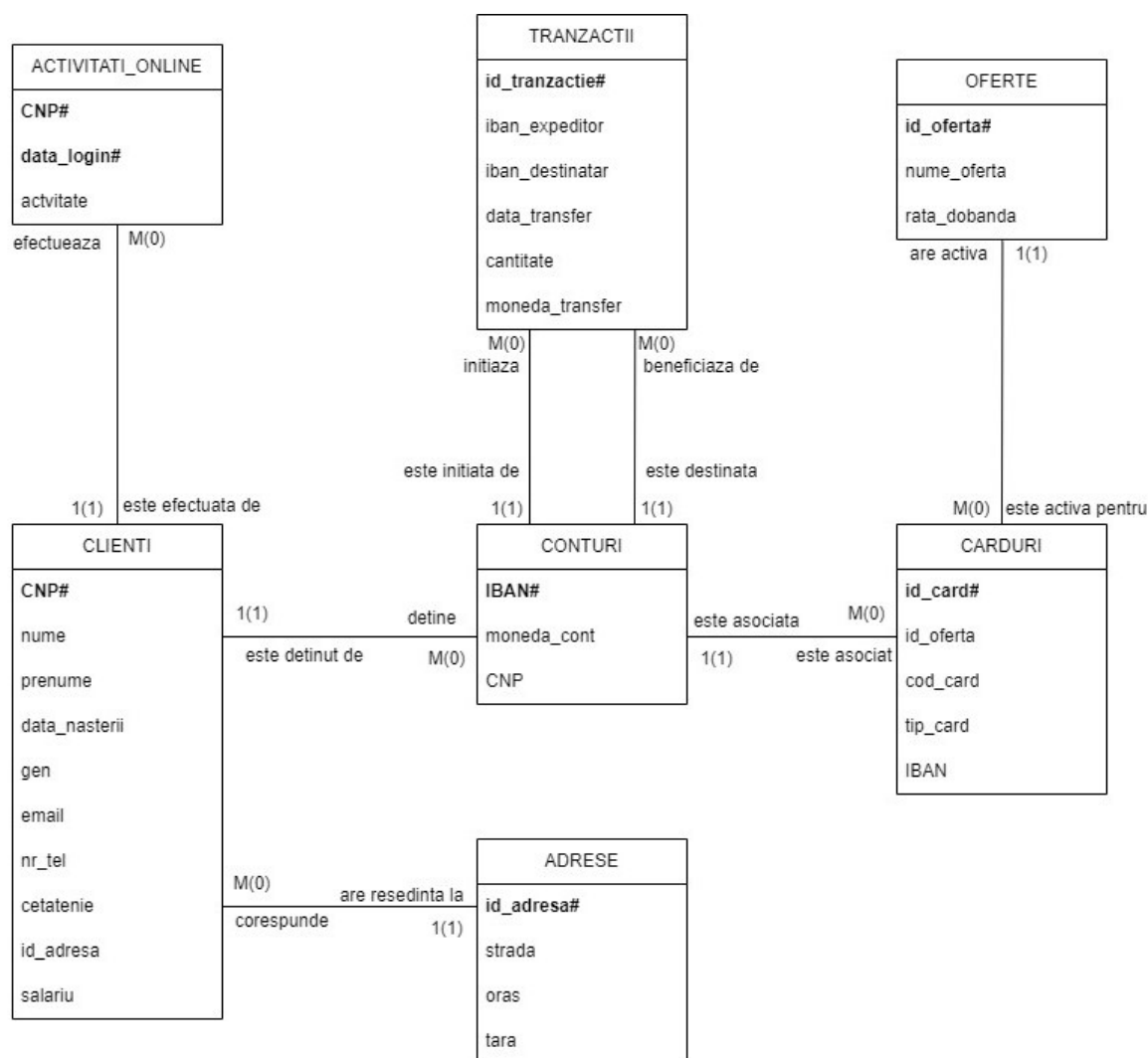


Figura 1: Digrama E-R aferentă proiectului

Descrierea entităților și atributelor

1. CLIENTI³ – este tabelul ce conține datele clienților băncii. Fiind persoane fizice, datele reținute includ informații precum CNP, numărul de telefon, genul, data nașterii etc.
 - a. CNP – VARCHAR(13) – identificatorul unic al oricărei înregistrări din tabel. Reprezintă codul numeric personal introdus conform decretului din 2 martie 1978 pentru cetățenii români și un alt fel de identificator (nr. id, nr. document etc.) pentru non-cetățeni.
 - b. NUME și PRENUME – VARCHAR(50) – numele și prenumele clientului.
 - c. DATA_NASTERII – DATE – data nașterii clientului, în format ISO (yyyy-mm-dd).
 - d. GEN – ENUM – genul clientului. Poate avea valorile ‘M’ sau ‘F’.
 - e. EMAIL – VARCHAR(55) – e-mail-ul clientului cu care poate fi contactat.
 - f. NR_TEL – VARCHAR(15) – numărul de telefon al clientului (fără prefix de țară).
 - g. CETĂȚENIE – ENUM – cetățenia clientului, din lista țărilor UE (format ISO-3166-1 alpha-3) sau ‘not_eu’ pentru țări în afara Uniunii.
 - h. ID_ADRESA – INT – id-ul adresei clientului, face legătura cu tabelul ADRESE.
 - i. SALARIU – INT – salariul clientului (dacă acesta este salariat).
2. ADRESE – tabelul ce conține diferite adrese din România și alte țări europene.
 - a. ID_ADRESA – INT – identificatorul unic al oricărei adrese din baza de date.
 - b. STRADA, ORAS – VARCHAR(40), VARCHAR(30) – reprezintă orașul și denumirea străzii din cadrul adresei.
 - c. TARA – ENUM – țara în care se află adresa (format ISO-3166-1 alpha-2 pentru țări UE sau ‘not_eu’ pentru țări ce nu fac parte din Uniune).
3. ACTIVITATI_ONLINE – tabelul ce conține istoricul activităților online din aplicația băncii, pentru fiecare client.
 - a. CNP – VARCHAR(13) – o parte din cheia primară compusă a tabelului, reprezintă CNP-ul clientului ce a efectuat o activitate
 - b. DATA_LOGIN – TIMESTAMP – reprezintă data și timpul exact în care o activitate a fost efectuată. Face parte din cheia primară compusă.
 - c. ACTIVITATE – ENUM – reprezintă activitatea efectuată de client (întrebare sold, schimbare setări etc.)

³ Orice asemănare de nume, CNP sau alte informații din baza de date folosită în proiect este pur întâmplătoare

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

4. CONTURI – tabelul ce conține informații despre conturile deschise de către clienți la bancă.
 - a. IBAN – VARCHAR(34) – indentificatorul unic al unui cont bancar. Are formatul specific României și conține identificadorul băncii ficționale (MBNK).
 - b. MONEDA_CONT – VARCHAR(3) – reprezintă moneda în care sunt exprimați banii din fiecare cont bancar.
 - c. CNP – VARCHAR(13) – face legătura cu tabelul CLIENTI, reprezintă CNP-ul (sau alternativele) clientului care deține contul.
5. TRANZACTII – tabelul ce reține datele despre orice tranzacție bancară efectuată între două conturi diferite.
 - a. ID_TRANZACTIE – INT – identificadorul unic pentru fiecare tranzacție
 - b. IBAN_EXPEDITOR, IBAN_DESTINATAR – VARCHAR(34) – reprezintă IBAN-urile conturilor ce participă la tranzacție. Cele 2 chei fac legătura cu 2 înregistrări diferite din tabelul CONTURI.
 - c. CANTITATE – INT – cantitatea de bani ce a fost transferată de la un cont la altul
 - d. MONEDA_TRANSFER – VARCHAR(3) – moneda în care s-a efectuat tranzacția. Nu depinde de moneda conturilor ce au participat la transfer.
6. CARDURI – tabelul ce conține informații despre cardurile bancare emise de bancă.
 - a. ID_CARD – INT – indentificatorul fiecărui card bancar emis.
 - b. ID_OFERTA – INT – identificadorul ofertei active pentru card. Face legătura cu tabelul OFERTE.
 - c. COD_CARD – VARCHAR(16) – conține numărul cardului (de 16 cifre, de pe fața cardului bancar).
 - d. TIP_CARD – ENUM – tipul cardului (debit sau credit).
 - e. IBAN – VARCHAR(34) – IBAN-ul contului asociat cardului bancar. Face legătura cu tabelul CONTURI.
7. OFERTE – tabelul ce conține informații despre ofertele valabile pentru un card bancar.
 - a. ID_OFERTA – INT – identificadorul unic pentru fiecare ofertă.
 - b. NUME_OFERTA – VARCHAR(30) – denumirea ofertei.
 - c. RATA_DOBANDA – DECIMAL(4, 2) – rata de dobândă asociată ofertei, exprimată în procente.

Descrierea relațiilor dintre entități

- **CLIENTI – ADRESE**
 - Un client locuiește la o singură adresă, considerată adresa curentă de reședință. Un client nu poate locui la mai mult de o adresă, și un client nu poate să nu aibă adresă. (minim 1, maxim 1 => un client are asociată doar o adresă, nu mai mult, nu mai puțin)
 - La o adresă pot locui mai mulți clienți în același timp. Deoarece adresa este una simplificată pentru siguranța clienților (se rețin doar țara, orașul, strada), exista mai mulți clienți ce au asociată aceeași adresă. De asemenea, pot exista adrese la care nu locuiește nimeni (salvate în baza de date deoarece a fost adresa unui fost client sau banca plănuiește să se extindă în zona respectivă). => (minim 0, maxim \forall)
- **CLIENTI – ACTIVITATI_ONLINE**
 - Un client poate efectua mai multe activități sau niciuna. (minim 0, maxim \forall)
 - O activitate este efectuată de un singur client. Nu poate exista o activitate ce nu a fost efectuată de nimeni sau o activitate ce a fost efectuată de mai mulți clienți. (minim 1, maxim 1 => o activitate este efectuată de exact un client)
- **CLIENTI – CONTURI**
 - Un client poate deschide oricâte conturi la bancă, dar este posibil și să nu aibă niciun cont deschis (în cazul în care cererea de cont nu a fost aprobată încă dar datele clientului au fost deja preluate). (minim 0, maxim \forall)
 - Un cont poate fi deținut de un singur client. Nu poate exista un cont fără deținător, sau un cont cu mai mulți deținători. (minim 1, maxim 1)
- **CONTURI – TRANZACTII**
 - un cont poate iniția sau beneficia de oricâte tranzacții sau niciuna. Conturile pot aparține aceluiași client sau unor clienți diferiți. (minim 0, maxim \forall).
 - o tranzacție este întotdeauna inițiată de un singur cont și destinată unui singur alt cont. Nu pot exista tranzacții destinate mai multor conturi în același timp, la fel cum mai multe conturi nu pot iniția aceeași tranzacție. (minim 1, maxim 1, atât pentru legătura cu expeditorul cât și pentru cea cu destinatarul)
- **CONTURI – CARDURI**
 - Un cont poate avea oricâte carduri bancare asociate, sau poate să nu aibă niciun card bancar emis. (minim 0, maxim \forall).
 - Un card bancar este emis pentru un singur cont, nu pot exista carduri bancare fără un cont asociat și nu pot exista carduri bancare asociate mai multor conturi deodată. (minim 1, maxim 1)

- **CARDURI – OFERTE**

- un card are o singură ofertă activă. Nu există carduri fără ofertă sau carduri cu mai multe oferte în același timp. (minim 1, maxim 1)
- o ofertă poate fi activă pentru mai multe carduri bancare în același timp sau pentru niciun card bancar (dacă este o ofertă introdusă foarte recent, de exemplu) . => (minim 0, maxim ∞)

Diagrama conceptuală

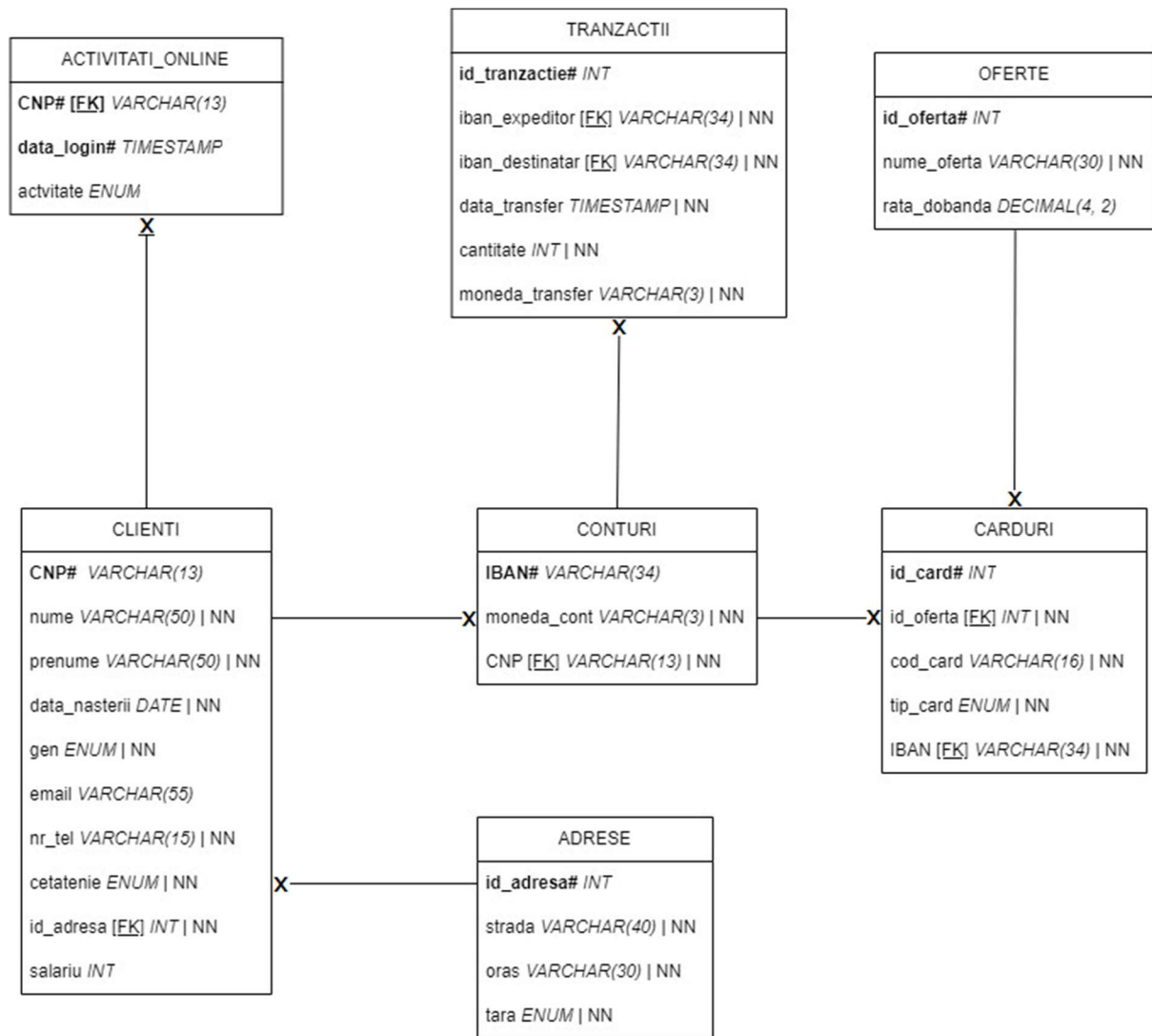


Figura 2: Diagrama conceptuală aferentă proiectului

Prezentarea constrângerilor de integritate

1. CLIENTI

- a. CNP – Primary Key, sunt impuse 2 constrângeri ce verifică validitatea câmpurilor GEN și DATA_NASTERII în funcție de CNP, pentru cetățenii români.
- b. NUME – Not Null – o persoană fizică fără un nume nu poate fi clientul băncii.
- c. PRENUME – Not Null – o persoană fizică fără un prenume nu poate fi clientul băncii.
- d. DATA_NASTERII – Not Null – un client trebuie să aibă o dată de naștere. Este impusă constrângerea de validare prin CNP (cifrele 2-8 din CNP corespund datei de naștere).
- e. GEN – Not Null – nu poate exista un client fără gen recunoscut legal. Este impusă constrângerea de validare prin CNP (prima cifră din CNP este 1/5 pentru bărbați și 2/6 pentru femei) pentru cetățeni români.
- f. EMAIL – Unique, este impusă o constrângere ce verifică structura validă a email-ului.
- g. NR_TEL – Not Null, Unique – clientul trebuie să aibă un număr de telefon pentru a putea fi contactat la nevoie de către un angajat al băncii.
- h. CETĂȚENIE – Not Null – orice client are o cetățenie.
- i. ID_ADRESA – Not Null – nu poate exista un client fără adresă.
- j. SALARIU – este impusă o constrângere ce garantează că salariul nu are o valoare negativă.

2. ADRESE

- a. ID_ADRESA – Primary Key.
- b. STRADA, ORAS, TARA – Not Null.

3. ACTIVITATI_ONLINE

- a. CNP – Primary Key, Foreign Key spre CLIENTI.CNP (on delete cascade).
- b. DATA_LOGIN – Primary Key.
- c. ACTIVITATE – nu are constrângeri impuse.

4. CONTURI

- a. IBAN – Primary Key, este impusă o constrângere ce verifică structura IBAN-ului (primele 2 caractere sunt 'RO', iar identificatorul băncii este 'MBNK').
- b. MONEDA_CONT – Not Null.
- c. CNP – Not Null, Foreign Key (face legătura cu CLIENTI.CNP) (on delete cascade).

5. TRANZACTII

- a. ID_TRANZACTIE – Primary Key.
- b. IBAN_EXPEDITOR, IBAN_DESTINATAR – Not Null, Foreign Keys (ambele fac legătura cu 2 înregistrări diferite din tabelul CONTURI). Este impusă constrângerea care garantează că cele 2 IBAN-uri sunt diferite.
- c. CANTITATE – Not Null, se verifică faptul că este pozitivă.
- d. MONEDA_TRANSFER – Not Null

6. CARDURI

- a. ID_CARD – Primary Key.
- b. ID_OFERTA – INT – identificatorul ofertei active pentru card. Face legătura cu tabelul OFERTE.
- c. COD_CARD – Not Null – nu poate exista un card fără cod de 16 cifre. Codul are exact 16 cifre, fapt verificat printr-o constrângere de tip CHECK (analog constrângerilor impuse asupra altor tabele).
- d. TIP_CARD – Not Null – orice card bancar emis este sau debit, sau credit.
- e. IBAN – Not Null, Foreign Key (legatură cu CONTURI.IBAN) (on delete cascade).

7. OFERTE – tabelul ce conține informații despre ofertele valabile pentru un card bancar.

- a. ID_OFERTA – Primary Key.
- b. NUME_OFERTA – Not Null – orice ofertă are o denumire.
- c. RATA_DOBANDA – se verifică faptul că valoarea ratei de dobândă este între 0 și 20 (%).

Schemele relaționale

Schemele relaționale ale modelului folosit în proiect sunt:

ADRESE(id_adresa#, strada, oras, tara);

CLIENTI(CNP#, nume, prenume, data_nasterii, gen, email, nr_tel, cetatenie, id_adresa [FK], salariu);

ACTIVITATI_ONLINE(CNP# [FK], data_login#, activitate);

CONTURI(IBAN#, moneda_cont, CNP [FK]);

OFERTE(id_oferta#, nume_oferta, rata_dobanda);

CARDURI(id_card#, id_oferta [FK], cod_card, tip_card, IBAN [FK]);

TRANZACTII(id_tranzactie#, iban_expeditor [FK], iban_destinatar [FK], data_transfer, cantitate, moneda_transfer);

IMPLEMENTAREA ÎN MYSQL

Crearea tabelelor și definirea constrângerilor ⁴

Tabelele au fost create în conformitate cu constrângerile lor, în ordinea prezentată mai jos.

```
CREATE TABLE `adrese` (  
    `id_adresa` int NOT NULL,  
    `strada` varchar(40) NOT NULL,  
    `oras` varchar(30) NOT NULL,  
    `tara` enum('AT', 'BE', 'BG', 'HR', 'CY', 'CZ', 'DK', 'EE',  
'FI', 'FR', 'DE', 'GR', 'HU', 'IE', 'IT', 'LV', 'LT', 'LU', 'MT',  
'NL', 'PL', 'PT', 'RO', 'SK', 'SI', 'ES', 'SE', 'NOT_EU') NOT NULL  
    DEFAULT 'RO',  
    PRIMARY KEY (`id_adresa`),  
    UNIQUE KEY `id_adresa_UNIQUE` (`id_adresa`),  
    CONSTRAINT `ID_VALID_AD` CHECK (`id_adresa` > 0)  
);
```

```
CREATE TABLE `clienti` (  
    `CNP` varchar(13) NOT NULL,  
    `nume` varchar(50) NOT NULL,  
    `prenume` varchar(50) NOT NULL,  
    `data_nasterii` date NOT NULL,  
    `gen` enum('M', 'F') NOT NULL,  
    `email` varchar(55) DEFAULT NULL,
```

⁴ Codul SQL pentru crearea tabelelor și constrângerilor, precum și cel pentru inserarea datelor poate fi găsit în arhiva primită, în fișierul Fota_Adrian_262_SQL.sql

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
`nr_tel` varchar(15) NOT NULL,
`cetatenie`
enum('AUT','BEL','BGR','HRV','CYP','CZE','DNK','EST','FIN','FRA','DEU',
',','GRC','HUN','IRL','ITA','LVA','LTU','LUX','MLT','NLD','POL','PRT',''
ROU','SVK','SVN','ESP','SWE','NOT_EU') DEFAULT 'ROU',
`id_adresa` int NOT NULL,
`salarium` int DEFAULT NULL,
PRIMARY KEY (`CNP`),
UNIQUE KEY `CNP_UNIQUE` (`CNP`),
UNIQUE KEY `nr_tel_UNIQUE` (`nr_tel`),
UNIQUE KEY `email_UNIQUE` (`email`),
KEY `FK_CLIENT_ADRESA` (`id_adresa`),
CONSTRAINT `FK_CLIENT_ADRESA` FOREIGN KEY (`id_adresa`)
REFERENCES `adrese` (`id_adresa`),
CONSTRAINT `GEN_VALID_CNP` CHECK (
    case
        when `cetatenie` <> 'ROU' then 'EXT'
        when `cetatenie` = 'ROU' then
            case
                when substr(`CNP`,1,1) in ('1','5') and
upper(`gen`) = 'M' then 'OK'
                when substr(`CNP`,1,1) in ('2','6') and
upper(`gen`) = 'F' then 'OK'
                else 'NO'
            end
        end in ('OK','EXT')
    ),
CONSTRAINT `NASTERE_VALIDA_CNP` CHECK (
    case
```

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
        when `cetatenie` <> 'ROU' then 'EXT'
        when `cetatenie` = 'ROU' then
            case
                when substr(`CNP`,2,6) =
date_format(`data_nasterii`, '%y%m%d') then 'OK'
                else 'NO'
            end
        end in ('OK','EXT')
    ),
    CONSTRAINT `NR_TEL_VALID` CHECK (length(`nr_tel`) > 5),
    CONSTRAINT `SALARIU_VALID` CHECK (`salariu` >= 0),
    CONSTRAINT `STRUCTURA_EMAIL_VALIDA` CHECK (`email` like '%%%.%')
) ;
```

```
CREATE TABLE `activitati_online` (
    `CNP` varchar(13) NOT NULL,
    `data_login` timestamp NOT NULL,
    `activitate`
enum('interogare_sold','contactare_suport','verificare_istoric','schi
mbare_setari') DEFAULT NULL,
    PRIMARY KEY (`CNP`,`data_login`),
    CONSTRAINT `FK_ACTIVITATI_CLIENT` FOREIGN KEY (`CNP`) REFERENCES
`clienti` (`CNP`) ON DELETE CASCADE ON UPDATE CASCADE
) ;
```

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
CREATE TABLE `conturi` (  
    `IBAN` varchar(34) NOT NULL,  
    `moneda_cont` varchar(3) NOT NULL,  
    `CNP` varchar(13) NOT NULL,  
    PRIMARY KEY (`IBAN`),  
    UNIQUE KEY `IBAN_UNIQUE` (`IBAN`),  
    KEY `fk_cont_client_ind` (`CNP`),  
    CONSTRAINT `FK_CONT_CLIENT` FOREIGN KEY (`CNP`)   
REFERENCES `clienti` (`CNP`) ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT `MYBANK_IBAN` CHECK (substr(`IBAN`,5,4) =  
'MBNK' and substr(`IBAN`,1,2) = 'RO')  
);
```

```
CREATE TABLE `oferte` (  
    `id_oferta` int NOT NULL,  
    `nume_oferta` varchar(30) NOT NULL,  
    `rata_dobanda` decimal(4,2) DEFAULT '0.00',  
    PRIMARY KEY (`id_oferta`),  
    UNIQUE KEY `id_oferta_UNIQUE` (`id_oferta`),  
    UNIQUE KEY `nume_oferta_UNIQUE` (`nume_oferta`),  
    CONSTRAINT `DOBANDA_VALIDA` CHECK (`rata_dobanda` >= 0 and  
`rata_dobanda` <= 20),  
    CONSTRAINT `ID_VALID_OF` CHECK (`id_oferta` > 0)  
);
```

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
CREATE TABLE `carduri` (  
    `id_card` int NOT NULL,  
    `id_oferta` int NOT NULL,  
    `cod_card` varchar(16) NOT NULL,  
    `tip_card` enum('debit','credit') NOT NULL,  
    `IBAN` varchar(34) NOT NULL,  
    PRIMARY KEY (`id_card`),  
    UNIQUE KEY `cod_card_UNIQUE` (`cod_card`),  
    KEY `FK_CARD_CONT` (`IBAN`),  
    KEY `FK_CARD_OFERTA` (`id_oferta`),  
    CONSTRAINT `FK_CARD_CONT` FOREIGN KEY (`IBAN`) REFERENCES  
`conturi` (`IBAN`) ON DELETE CASCADE ON UPDATE CASCADE,  
    CONSTRAINT `FK_CARD_OFERTA` FOREIGN KEY (`id_oferta`) REFERENCES  
`oferte` (`id_oferta`),  
    CONSTRAINT `CARD_CODE_LENGTH` CHECK (char_length(`cod_card`)  
= 16),  
    CONSTRAINT `ID_CARD_VALID` CHECK (`id_card` > 0)  
);
```

```
CREATE TABLE `tranzactii` (  
    `id_tranzactie` int NOT NULL,  
    `iban_expeditor` varchar(34) NOT NULL,  
    `iban_destinatar` varchar(34) NOT NULL,  
    `data_transfer` timestamp NOT NULL,  
    `cantitate` int NOT NULL,  
    `moneda_transfer` varchar(3) NOT NULL,  
    PRIMARY KEY (`id_tranzactie`),  
    KEY `FK_CONT_DESTINATAR` (`iban_destinatar`),  
    KEY `FK_CONT_EXPEDITOR` (`iban_expeditor`),
```


UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
        CONSTRAINT `FK_CONT_DESTINATAR` FOREIGN KEY
(`iban_destinatar`) REFERENCES `conturi` (`IBAN`),

        CONSTRAINT `FK_CONT_EXPEDITOR` FOREIGN KEY
(`iban_expeditor`) REFERENCES `conturi` (`IBAN`),

        CONSTRAINT `CANTITATE_VALIDA` CHECK ((`cantitate` > 0)),

        CONSTRAINT `DIFF_IBAN` CHECK
(strcmp(`iban_expeditor`, `iban_destinatar`) <> 0),

        CONSTRAINT `ID_VALID_TRANZ` CHECK (`id_tranzactie` > 0)
);
```

Inserarea datelor ⁵

```
INSERT INTO ADRESE VALUES(1, "Piscotului", "Bucuresti", "RO");
INSERT INTO ADRESE VALUES(76, "Maslinului", "Timisoara", "RO");
INSERT INTO ADRESE VALUES(165, "Tineretului", "Onesti", "RO");
INSERT INTO ADRESE VALUES(217, "Bartok Bela", "Budapesta", "HU");
INSERT INTO CLIENTI VALUES("1531001485753", "Barbu", "Anton", "1953-10-01", "M", NULL, "0745076142", "ROU", 158, NULL);
INSERT INTO CLIENTI VALUES("1771216220833", "Fieraru", "Remus", "1977-12-16", "M", "fieraru.remus@outlook.com", "0767450827", "ROU", 88, 10100);
INSERT INTO CLIENTI VALUES("2750316082960", "Bursuc", "Ileana", "1975-03-16", "F", "ileana.bursuc@gmail.com", "0734900982", "ROU", 66, 8000);
INSERT INTO CLIENTI VALUES("H90176599", "Politoglou", "Michalis", "1981-02-20", "M", "michapolitoglou@gmail.com", "0225719487", "GRC", 124, 7800);
INSERT INTO ACTIVITATI_ONLINE VALUES("1690612392587", "2023-12-25 17:11:37", "contactare_suport");
```

⁵ În documentație vor fi prezentate doar câteva exemple de inserare a datelor pentru fiecare tabel. Codul complet cu inserarea tuturor înregistrărilor poate fi găsit în fișierul .sql din arhivă.

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
INSERT INTO ACTIVITATI_ONLINE VALUES("2870228199148", "2023-12-25
17:12:24", "verificare_istoric");

INSERT INTO ACTIVITATI_ONLINE VALUES("2911224168862", "2023-12-25
17:23:31", "schimbare_setari");

INSERT INTO ACTIVITATI_ONLINE VALUES("5041004362347", "2023-12-25
17:22:52", "contactare_suport");

INSERT INTO CONTURI VALUES("RO09MBNKRON87ZLC3Y4FQLL2", "RON",
"5051030101299");

INSERT INTO CONTURI VALUES("RO09MBNKUSDZ2Q2NQI3CF62O", "USD",
"1620414516332");

INSERT INTO CONTURI VALUES("RO13MBNKRON4N05824S1S5JC", "RON",
"5010827396429");

INSERT INTO CONTURI VALUES("RO15MBNKUSDGYXFD59VAVM98", "USD",
"1770609255111");

INSERT INTO OFERTE VALUES(1, "Standard RO", 1);
INSERT INTO OFERTE VALUES(3, "Gold RO", 2);
INSERT INTO OFERTE VALUES(5, "Gold Star", 3);
INSERT INTO OFERTE VALUES(9, "Platinum Star", 5.5);

INSERT INTO CARDURI VALUES(4, 5, "9754986537624342", "debit",
"RO27MBNKRONU9YMIOIOI7OLZ");

INSERT INTO CARDURI VALUES(36, 4, "8759044374359868", "debit",
"RO46MBNKRON5V9N9FH5P9EV9");

INSERT INTO CARDURI VALUES(46, 2, "6451791174881882", "debit",
"RO60MBNKRONR69S3YF0BV290");

INSERT INTO CARDURI VALUES(55, 5, "2643392244285420", "credit",
"RO71MBNKRONDVCQVKX7DPRK2");

INSERT INTO TRANZACTII VALUES(6, "RO49MBNKUSD TVMR6WTJ0CKZX",
"RO23MBNKRONMHSJJSQUMUBJ0", "2023-12-25 21:15:30", 25, "USD");

INSERT INTO TRANZACTII VALUES(10, "RO49MBNKRON SCHC8BW5QZOF8",
"RO01MBNKRONJNORRQSSSEUO7", "2023-12-28 19:52:24", 2600, "USD");

INSERT INTO TRANZACTII VALUES(23, "RO60MBNKRONR69S3YF0BV290",
"RO66MBNKUSD885UDR298QM8H", "2023-12-11 17:59:16", 4600, "RON");
```

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

```
INSERT INTO TRANZACTII VALUES (30, "RO50MBNKUSDJ91F9NFTTLOZV",  
"RO75MBNKRON6R0APAXRAFI4", "2023-12-10 16:38:51", 50, "USD");
```

The screenshot shows MySQL Workbench with two queries and their results.

Query 1 (Left):

```
SELECT CL.CNP, CL.NUME, CL.PRENUME, AD.ORAS, CL.SALARIU  
FROM CLIENTI CL  
JOIN ADRESE AD ON CL.ID_ADRESA = AD.ID_ADRESA  
ORDER BY CL.SALARIU DESC;
```

Result Grid 1 (Left):

CNP	NUME	PRENUME	ORAS	SALARIU
1950908192156	Popescu	Mircea	Tulcea	13400
2700728442083	Tataru	Sabina	Cluj-Napoca	13200
0086GD50	Halasz	Janos	Budapesta	13000
1550610118834	Stancu	Adrian	Bacau	13000
2881018410338	Predoiu	Ligia	Paris	12600
1970419320579	Oprea	Aurel	Deva	12400
1960228489327	Codreanu	Stefan	Medgidia	12300
1901031482215	Cojocaru	Marcel	Tulcea	12200
174780184F	Paez	Adara	Barcelona	12000

Query 2 (Right):

```
SELECT * FROM TRANZACTII  
WHERE CANTITATE > 1000 AND MONEDA_TRANSFER = "RON";
```

Result Grid 2 (Right):

id_tranzactie	iban_expositor	iban_destinatar	data_transfer	cantitate	moneda_transfer
1	RO39MBNKRONBCSIR2LQHAT86	RO93MBNKRONOD09QOXDPNIBU	2023-12-24 22:16:10	2600	RON
2	RO23MBNKRON94XY692JIO135	RO50MBNKUSDJ91F9NFTTLOZV	2023-12-24 22:22:22	5130	RON
9	RO87MBNKUSD5EWM9LC4BA8215	RO60MBNKRONUK8EJQJDFZBU8	2023-12-27 21:51:49	8300	RON
16	RO47MBNKUSDJSDBR8R17DXMO	RO83MBNKUSDYJLXJGQ3NXAS5	2023-12-09 19:54:53	2300	RON
17	RO23MBNKRONMHSHJ5QUMJB30	RO41MBNKRONDTLJR9KMKTE4	2023-12-16 12:14:05	2500	RON
23	RO60MBNKRONR69S3YF0BV290	RO66MBNKUSD889UDR298QM8H	2023-12-10 19:59:16	4600	RON
27	RO08MBNKRON217DCT3NG8K10	RO75MBNKEURVQZ5UPD114K1	2023-12-12 22:23:53	6000	RON
32	RO80MBNKRON73OOCK98WAD21	RO02MBNKUSD65S1220T88076	2023-12-15 23:15:47	4000	RON
39	RO88MBNKUSD0EV373HH1NF1M	RO98MBNKRONUTGDZPQ32C5V	2023-12-03 15:34:12	3600	RON
40	RO08MBNKUSDJT2550VMT037C	RO75MBNKRON6R0APAXRAFI4	2023-12-24 18:32:52	1200	RON

Figura 3: Exemplificarea unor query-uri simple după crearea tabelor și introducerea datelor

INCHEIERE

Concluzii

În finalizarea proiectului acesta, am reușit să modelez și să implementez în MySQL o bază de date simplă din domeniul bancar. Am învățat să lucrez cu un SGBD nou, utilizând IDE-ul său specific, MySQL Workbench. Experiența lucrării asupra proiectului mi-a oferit o mai bună înțelegere a funcționării bazelor de date relaționale – precum a modelării legăturilor între entități și a definirii constrângerilor. În final, proiectul nu reprezintă doar un set de coduri și tabele, ci o experiență formativă care a contribuit la dezvoltarea mea personală și academică, dar și la îmbogățirea cunoștințelor mele în gestionarea bazelor de date.

Bibliografie

- <https://www.mysql.com/products/workbench/>
- <https://www.oracle.com/mysql/what-is-mysql/>