



**UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI**

DISCIPLINA: BAZE DE DATE

**SISTEM DE GESTIONARE A
ÎNCHIRIERILOR PENTRU UN
MAGAZIN DE BICICLETE**

**Coordonator,
ș.l. dr.ing
Mironeanu Cătălin**

**Student,
Chiriac Gabriel
Grupa 1311B**

Iași, 2021

Etapa intai:

În această etapă, a trebuit să modelăm proiectul prin alegerea temei și formularea unei descrieri non-tehnice care să explice în termeni accesibili contextul și funcționalitățile principale ale acestuia. Am detaliat scopul proiectului, nevoile pe care le satisface și modul în care acesta va funcționa, fără a utiliza termeni tehnici sau specifici bazelor de date. Această descriere a servit drept punct de plecare pentru analiza ulterioară.

Pe baza descrierii non-tehnice, am dezvoltat o descriere tehnică, care a inclus identificarea entităților, atributelor și relațiilor necesare pentru modelarea bazei de date. Această analiză tehnică a fost esențială pentru a transforma cerințele generale în structuri concrete de baze de date, pregătind astfel proiectul pentru etapele următoare.

Descrierea proiectului: Sistem de gestionare a închirierilor pentru un magazin de biciclete

Proiectul modelează activitatea unui magazin care închiriază biciclete și accesorii clienților săi. Sistemul permite înregistrarea clienților, gestionarea contactelor acestora, administrarea stocului de biciclete și accesorii, precum și calcularea costurilor închirierilor.

Entități principale:

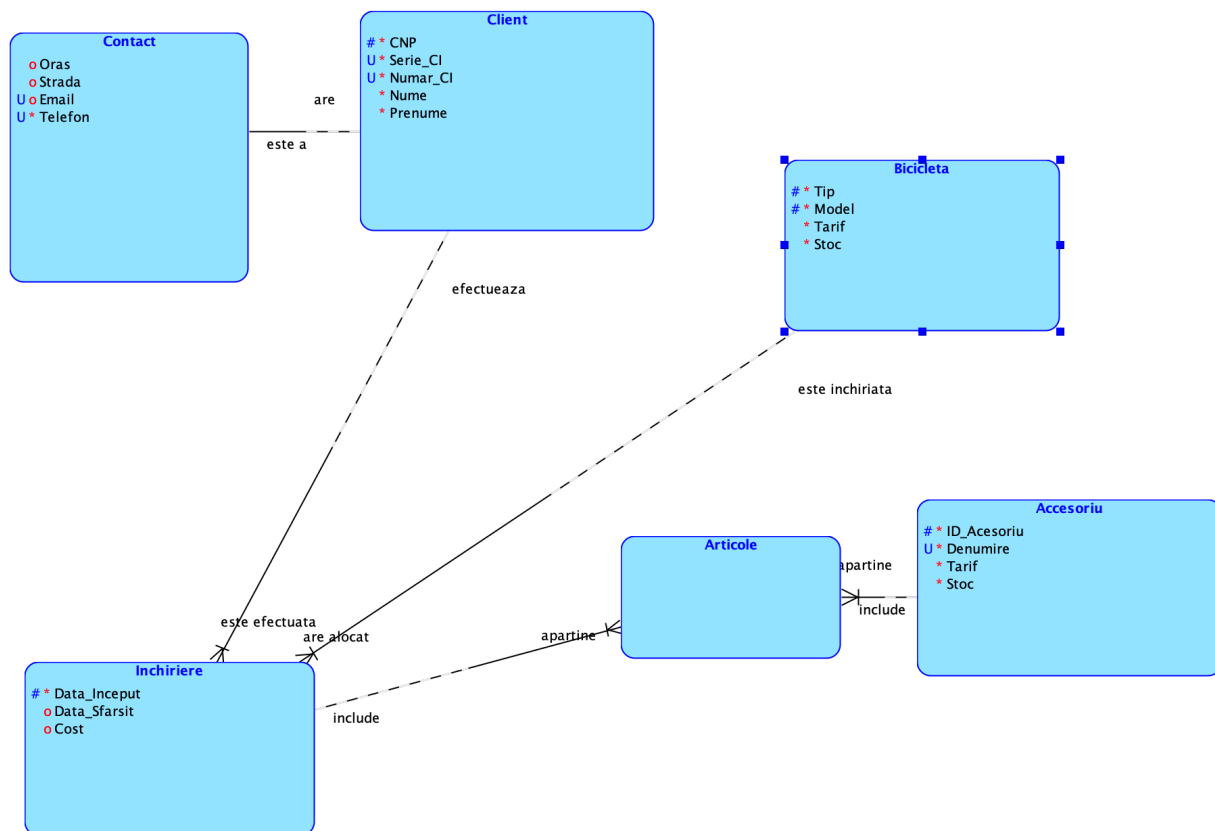
1. **Client**
 - Informații de bază despre clienți.
 - Atribute: CNP, Serie_CI, Numar_CI, Nume, Prenume.
2. **Contact**
 - Informații de contact ale clienților, gestionate separat pentru o organizare mai clară.
 - Atribute: CNP (cheie externă din Client), Oras, Strada, Email, Telefon.
3. **Bicicletă**
 - Stocul și caracteristicile bicicletelor disponibile pentru închiriere.
 - Atribute: Tip (electrică/oras/gravel/mountain/cursieră), Model (bărbați/dame/copii), Tarif, Stoc.
4. **Închiriere**
 - Informații despre închirierile realizate de clienți, inclusiv calculul costurilor în funcție de perioada și tipul bicicletei.
 - Atribute: CNP (cheie externă din Client), Tip și Model (chei externe din Bicicletă), Data_Început, Data_Sfârșit, Cost_Total.
5. **Accesorii**
 - Accesorii disponibile pentru închiriere împreună cu bicicletele.
 - Atribute: ID_Accesoriu, Denumire, Tarif, Stoc.
6. **Articole**
 - Relația dintre închiriere și accesorii, gestionând accesoriiile asociate unei închirieri specifice.
 - Atribute: ID_Închiriere (cheie externă din Închiriere), ID_Accesoriu (cheie externă din Accesorii), Cost_Accesoriu.

Funcționalitatea proiectului:

1. Magazinul are un stoc inițial de 150 de biciclete, distribuite în cele 5 tipuri (electrică, oraș, gravel, mountain, cursieră) și cele 3 modele (bărbați, dame, copii).
2. Clienții pot închiria biciclete pentru una sau mai multe zile, iar costul se calculează în funcție de numărul de zile și tipul bicicletei.
3. Închirierile sunt gestionate doar pentru clienții deja înregistrați în sistem.
4. Magazinul oferă accesorii suplimentare (de exemplu: cască, claxon, lumini), iar costul lor este adăugat la cel al bicicletei

Etapa a doua:

Pe baza descrierii realizate în prima etapă, am reușit să dezvolt următoarea diagramă logică utilizând Data Modeler.



1. Tabelul client

- **CNP (CHAR(13))**: Cheia primară. Reprezintă un identificator unic pentru fiecare client, bazat pe codul numeric personal. Formatul de 13 caractere este standard pentru România.
- **SERIE_CI (CHAR(2))**: Seria cărții de identitate a clientului. Lungimea fixă de 2 caractere reflectă structura standard.
- **NUMAR_CI (NUMBER(6))**: Numărul cărții de identitate, stocat ca număr cu 6 cifre.
- **NUME (VARCHAR2(20))**: Numele clientului. Lungimea de 20 de caractere permite stocarea numelor uzuale.
- **PRENUME (VARCHAR2(50))**: Prenumele clientului. Dimensiunea de 50 de caractere este adecvată pentru prenume mai lungi.

2. Tabelul contact

- **CNP (CHAR(13))**: Cheie străină referențiată din tabelul **client**. Indică persoana asociată informațiilor de contact.
- **ORAS (VARCHAR2(15))**: Orașul de domiciliu al clientului. Lungimea de 15 caractere este suficientă pentru majoritatea numelor de orașe.
- **STRADA (VARCHAR2(30))**: Strada pe care locuiește clientul. Lungimea de 30 permite descrieri mai detaliate.
- **EMAIL (VARCHAR2(20))**: Adresa de email a clientului. Dimensiunea de 20 este potrivită pentru formate uzuale.
- **TELEFON (CHAR(10))**: Numărul de telefon al clientului, format din 10 cifre, conform standardului românesc.

3. Tabelul bicicleta

- **TIP (VARCHAR2(9))**: Parte a cheii primare. Specifică categoria bicicletei (ex: electrica, gravel, oras, mountain, cursiera). Lungimea de 9 caractere este suficientă pentru clasificări clare.
- **MODEL (VARCHAR2(7))**: Parte a cheii primare. Specifică mărimea bicicletei (ex: barbati, dame, copii).
- **TARIF (NUMBER(3))**: Costul închirierii unei biciclete pentru o zi. Valoarea maximă permisă este 999.
- **STOC (NUMBER(2))**: Numărul de biciclete disponibile din fiecare tip și model, limitat la 99.

4. Tabelul inchiriere

- **DATA_INCEPUT (DATE)**: Data la care a început închirierea bicicletei.
- **DATA_SFARSIT (DATE)**: Data la care închirierea s-a încheiat. Este opțională, deoarece perioada poate fi nedeterminată.
- **COST (NUMBER(6))**: Costul total al închirierii, calculat pe baza tipului bicicletei și a numărului de zile. Limita maximă este 999.999.
- **CNP (CHAR(13))**: Cheie străină referențiată din tabelul **client**. Indică persoana care a realizat închirierea.

- **TIP (VARCHAR2(9))**: Cheie străină referențiată din tabelul **bicicleta**. Specifică tipul bicicletei închiriate.
- **MODEL (VARCHAR2(7))**: Cheie străină referențiată din tabelul **bicicleta**. Specifică modelul bicicletei închiriate.

5. Tabelul accesorii

- **ID_ACESORIU (NUMBER(5))**: Cheia primară. Poate stoca până la 99.999 accesorii unice.
- **DENUMIRE (VARCHAR2(30))**: Numele accesoriului (ex: Casca, Claxon). Lungimea de 30 permite descrieri detaliate.
- **TARIF (NUMBER(3))**: Costul pentru închirierea accesoriului, exprimat în unități monetare. Limita maximă este 999.
- **STOC (NUMBER(2))**: Numărul maxim de accesorii disponibile în stoc, limitat la 99.

6. Tabelul articole

- **CNP (CHAR(13))**: Cheie străină referențiată din tabelul **client**. Indică persoana care a închiriat accesoriul.
- **DATA_INCEPUT (DATE)**: Data la care accesoriul a fost inclus în închiriere.
- **TIP (VARCHAR2(9))**: Tipul bicicletei asociate închirierii (ex: mountain, oras).
- **MODEL (VARCHAR2(7))**: Modelul bicicletei (ex: barbati, dame, copii).
- **ID_ACESORIU (NUMBER(5))**: Cheie străină referențiată din tabelul **accesoriu**. Indică accesoriul închiriat.

Tot în această etapă, am prevăzut crearea constrângerilor necesare pentru asigurarea integrității datelor, precum **primary key**, **unique**, **not null** și **check**.

1. Cheile Primare

- **Tabela client**
 - **client_pk**: Definește cnp drept cheia primară. Aceasta asigură unicitatea fiecărui client, fiecare având un CNP unic, conform standardelor din România.
- **Tabela contact**
 - **contact_pk**: Definește cnp drept cheia primară, indicând că fiecare client poate avea un singur set de date de contact asociat.
- **Tabela bicicleta**
 - **bicicleta_pk**: Definește combinația tip și model drept cheia primară, asigurând unicitatea fiecărui tip și model de bicicletă.
- **Tabela inchiriere**
 - **inchiriere_pk**: Definește combinația cnp, data_inceput, tip, model drept cheia primară, ceea ce înseamnă că o combinație unică de client, bicicletă și dată de începere a închirierii este obligatorie.
- **Tabela accesorii**
 - **accesoriu_pk**: Definește id_acesoriu drept cheia primară, ceea ce garantează unicitatea fiecărui accesoriu.

- **Tabela articole**
 - **articole_pk**: Definește combinația cnp, data_inceput, tip, model, id_acesoriu drept cheie primară, ceea ce asigură unicitatea fiecărui accesoriu închiriat pentru un client.

2. Constrângerile de Unicitate (Unique)

- **Tabela client**
 - **client_serie_ci_numar_ci_un**: Asigură unicitatea perechii serie_ci și numar_ci, astfel încât un set de date de identificare să fie asociat unui singur client.
- **Tabela contact**
 - **contact_email_un**: Asigură unicitatea adreselor de email, evitând înregistrări duplicate.
 - **contact_telefon_un**: Asigură unicitatea numerelor de telefon.
- **Tabela accesoriu**
 - **acesoriu_denumire_un**: Asigură unicitatea denumirii fiecărui accesoriu, prevenind intrări duplicate cu același nume.

3. Constrângerile NOT NULL

- **Tabela client**
 - Toate coloanele (cnp, serie_ci, numar_ci, nume, prenume) sunt obligatorii pentru a completa o înregistrare validă.
- **Tabela contact**
 - telefon și cnp sunt obligatorii pentru fiecare contact, deoarece sunt necesare pentru identificare și comunicare.
- **Tabela bicicleta**
 - Toate coloanele (tip, model, tarif, stoc) sunt obligatorii, deoarece sunt esențiale pentru identificarea și gestionarea bicicletelor.
- **Tabela inchiriere**
 - Coloanele data_inceput, cnp, tip, model sunt obligatorii pentru a completa o închiriere.
- **Tabela accesoriu**
 - Toate coloanele (id_acesoriu, denumire, tarif, stoc) sunt obligatorii pentru o înregistrare validă.
- **Tabela articole**
 - Toate coloanele (cnp, data_inceput, tip, model, id_acesoriu) sunt obligatorii pentru a înregistra un articol.

4. Constrângerile de Sintaxă (Check)

- **Tabela client**
 - **client_cnp_ck**: Verifică formatul CNP-ului conform standardelor românești.
 - **client_serie_ci_ck**: Permite doar seriile XT și ZT.
 - **client_numar_ci_ck**: Asigură că numărul CI este strict pozitiv.

- **client_nume_ck** și **client_prenume_ck**: Verifică dacă numele și prenumele conțin doar caractere valide.
- **Tabela contact**
 - **contact_oras_ck**: Permite doar caractere alfanumerice, spații și cratime pentru oras.
 - **contact_email_ck**: Verifică dacă emailul este într-un format valid.
 - **contact_telefon_ck**: Permite doar numere de telefon românești care încep cu 07.
- **Tabela bicicleta**
 - **bicicleta_tip_ck**: Permite doar tipuri predefinite de biciclete (ex. mountain, cursiera).
 - **bicicleta_model_ck**: Permite doar modelele barbati, dame, copii.
 - **bicicleta_tarif_ck**: Asigură că tariful este una dintre valorile predefinite.
 - **bicicleta_stoc_ck**: Verifică dacă stocul este între 0 și 10.
- **Tabela inchiriere**
 - **inchiriere_data_sfarsit_ck**: Verifică dacă data_sfarsit este mai mare sau egală cu data_inceput.
- **Tabela accesoriu**
 - **accesoriu_denumire_ck**: Permite doar denumiri predefinite pentru accesorii (ex. Casca, Claxon).
 - **accesoriu_tarif_ck**: Permite doar valorile 25, 50, 75, 100 pentru tarif.
 - **accesoriu_stoc_ck**: Asigură că stocul este mai mare sau egal cu 0.

Etapa a treia:

În această etapă, am avut ca sarcină principală definirea relațiilor dintre entitățile din modelul logic al bazei de date.

1. **Relația dintre Client și Contact:**
 - Fiecare client are un set unic de informații de contact. Relația este de tip **1:1**, deoarece fiecare client poate avea un singur set de detalii de contact.
2. **Relația dintre Client și Închiriere:**
 - Fiecare client poate efectua una sau mai multe închirieri, iar fiecare închiriere este asociată unui singur client. Relația este de tip **1:M**.
3. **Relația dintre Închiriere și Bicicletă:**
 - Fiecare închiriere este asociată cu un anumit tip și model de bicicletă. Relația este de tip **M:1**, deoarece mai multe închirieri pot implica același tip și model de bicicletă.
4. **Relația dintre Închiriere și Articole:**
 - Închirierile pot include unul sau mai multe articole (accesorii). Relația este de tip **1:M**, unde o închiriere poate avea mai multe articole asociate.
5. **Relația dintre Articole și Accesorii:**
 - Fiecare articol dintr-o închiriere corespunde unui accesoriu specific. Relația este de tip **M:1**, deoarece mai multe articole pot fi asociate cu același tip de accesoriu.

Spargerea relației M:N dintre Închiriere și Accesoriu

- În modelul inițial, relația dintre Închiriere și Accesoriu era de tip **M:N**, ceea ce însemna că o închiriere putea include mai multe accesorii și un accesoriu putea fi asociat cu mai multe închirieri.
- Pentru a respecta cerințele modelării corecte a bazei de date, această relație a fost spartă prin introducerea entității intermediare Articole.
 - Articole conține detalii despre accesorii asociate fiecărei închirieri, stabilind două relații 1:M:
 - Între Închiriere și Articole (1:M).
 - Între Articole și Accesoriu (M:1).

Această modificare permite o mai bună organizare a datelor și facilitează gestionarea accesoriilor asociate fiecărei închirieri.

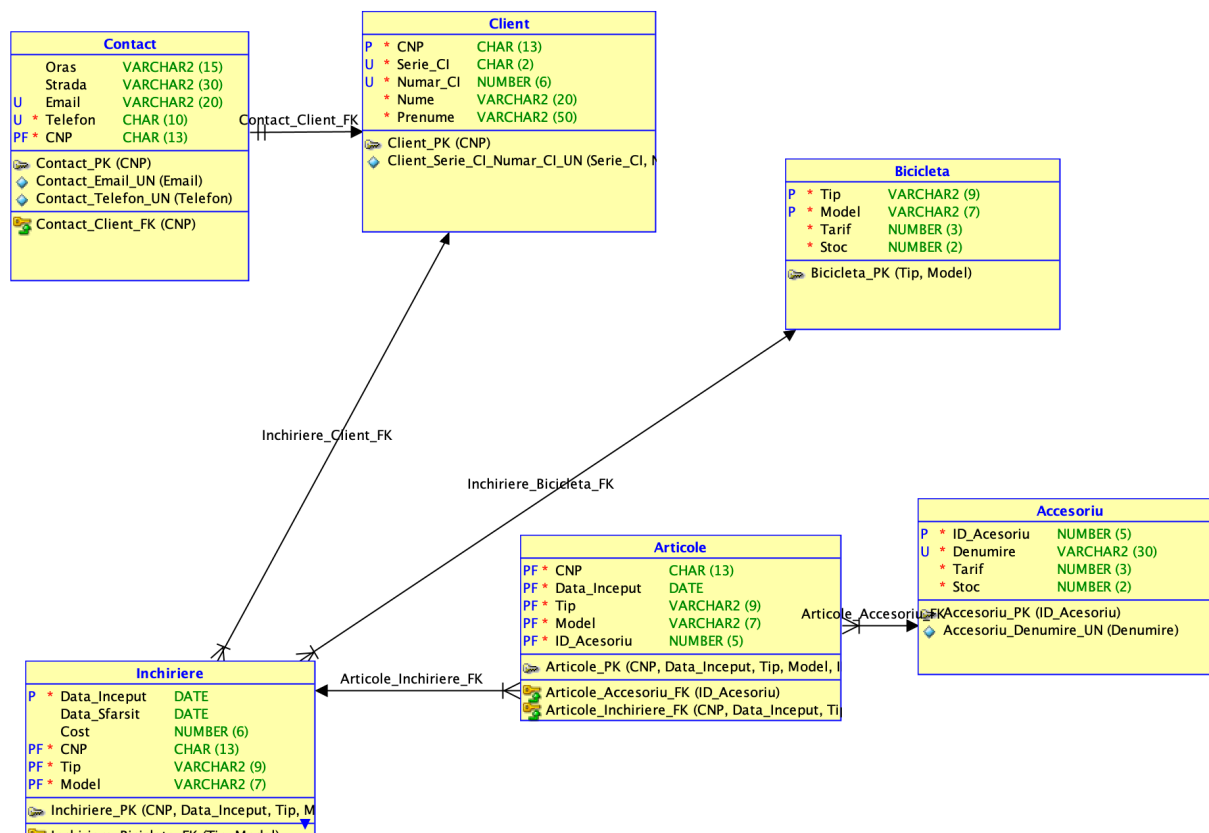
Pentru că am stabilit relațiile între entități, a fost necesar să definim și **chei externe (Foreign Keys)**, asigurând legătura logică între tabele, acestea permițând conectarea entităților.

Cheile Străine (Foreign Keys)

- **Tabela contact**
 - **contact_client_fk**: Leagă cnp din contact de cnp din client, asigurând că datele de contact există doar pentru clienți validați.
- **Tabela inchiriere**
 - **inchiriere_client_fk**: Leagă cnp din inchiriere de cnp din client, asigurând că o închiriere este realizată doar de clienți validați.
 - **inchiriere_bicicleta_fk**:
Leagă tip și model din inchiriere de tip și model din bicicleta, asigurând că bicicletele închiriate există în stoc.
- **Tabela articole**
 - **articole_inchiriere_fk**: Leagă cnp, data_inceput, tip, model din articole de aceleași coloane din inchiriere, asigurând că articolele închiriate sunt legate de o închiriere validă.
 - **articole_accesoriu_fk**:
Leagă id_accesoriu din articole de id_accesoriu din accesoriu, asigurând că accesoriiile închiriate există.

Etapă a patra:

În această etapă, am început prin generarea Modelului Relational, rezultând următoarea diagramă:



Ulterior, am adăugat triggerre pentru automatizarea unor operații, precum generarea valorilor pentru coloanele cu auto increment sau verificarea datelor introduse. Aceste triggerre au fost incluse după generarea modelului relational, pentru a extinde funcționalitatea și a asigura integritatea datelor.

1. Trigger pentru tabelul Inchiriere - Trg_Inchiriere_Data

- **Scop:** Verifică dacă data de început a unei închirieri (Data_Inceput) este validă, adică mai mare decât data curentă. Astfel, se previne inserarea sau actualizarea unor închirieri cu date invalide.
- **Funcționalitate:** Dacă Data_Inceput este mai mică sau egală cu data curentă (SYSDATE), se generează o eroare cu un mesaj clar pentru utilizator prin utilizarea funcției RAISE_APPLICATION_ERROR.

2. Trigger pentru tabelul Accesorii - accesoriu_id_acesoriu_trg

- **Scop:** Automatizează generarea valorii pentru coloana id_acesoriu atunci când aceasta nu este specificată la inserare.
- **Funcționalitate:** Utilizează secvența accesoriu_id_acesoriu_seq pentru a genera automat un ID unic pentru fiecare accesoriu nou adăugat în tabel. Secvența începe de la valoarea 1000 și crește cu un pas de 37.

Etapa a cincea:

În sfârșit, după toate etapele parcurse, am ajuns să generăm codul de creare al bazei de date utilizând modelul relațional definit în Oracle Data Modeler. Acesta se afla în fișierul ***script_creare.ddl***.

Pe lângă generarea codului de creare, am realizat și un set de teste pentru verificarea funcționalității bazei de date dar și a constrângerilor, documentate în fișierul ***script_inserturi.sql***.

Pe lângă generarea codului de creare și testare, am creat două scripturi esențiale:

1. ***populare_tabele.sql***: Acest script se ocupă de popularea tabelelor cu date inițiale necesare pentru a simula structura bazei de date. Aceste date sunt esențiale pentru a permite gestionarea unei baze de date funcționale. De exemplu, în acest script se introduc date în tablele bicicleta si accesoriu, pregătind astfel baza de date cu stocul magazinului.
2. ***inserare_client.sql***: Scriptul simulează procesul de închiriere, pornind de la adăugarea unui client nou și continuând cu gestionarea închirierilor propriu-zise. Prin acest script:
 - Se inserează date despre client, contact și informațiile necesare pentru închiriere.
 - Se calculează costul închirierii și se înregistrează în tablele.